12-2-2018

Multi-Modal Deep Learning to Understand Vision and Language

Shagan Sah
sxs4337@rit.edu

Multi-Modal Deep Learning to Understand Vision and Language

by

Shagan Sah

B.E. University of Pune, 2009
M.S. Rochester Institute of Technology, 2013

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Chester F. Carlson Center for Imaging Science
College of Science
Rochester Institute of Technology

December 2, 2018

Signature of the Author _____

Accepted by _____
Coordinator, Ph.D. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

Ph.D. DEGREE DISSERTATION

The Ph.D. Degree Dissertation of Shagan Sah
has been examined and approved by the
dissertation committee as satisfactory for the
dissertation required for the
Ph.D. degree in Imaging Science

Dr. Raymond Ptucha, Dissertation Advisor Date

Coordinator Ph.D. Degree Program Date

Dr. Carl Salvaggio Date

Dr. Nathan Cahill Date

Dr. Leon Reznik Date

DISSERTATION RELEASE PERMISSION
ROCHESTER INSTITUTE OF TECHNOLOGY
COLLEGE OF SCIENCE
CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE

Title of Dissertation:

Multi-Modal Deep Learning to Understand Vision and Language

I, Shagan Sah, hereby grant permission to Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature _____

Date

Multi-Modal Deep Learning to Understand Vision and Language

by

Shagan Sah

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Doctor of Philosophy Degree
at the Rochester Institute of Technology

Abstract

Developing intelligent agents that can perceive and understand the rich visual world around us has been a long-standing goal in the field of artificial intelligence. In the last few years, significant progress has been made towards this goal and deep learning has been attributed to recent incredible advances in general visual and language understanding. Convolutional neural networks have been used to learn image representations while recurrent neural networks have demonstrated the ability to generate text from visual stimuli. In this thesis, we develop methods and techniques using hybrid convolutional and recurrent neural network architectures that connect visual data and natural language utterances.

Towards appreciating these methods, this work is divided into two broad groups. Firstly, we introduce a general purpose attention mechanism modeled using a continuous function for video understanding. The use of an attention based hierarchical approach along with automatic boundary detection advances state-of-the-art video captioning results. We also develop techniques for summarizing and annotating long videos. In the second part, we introduce architectures along with training techniques to produce a common connection space where natural language sentences are efficiently and accurately connected with visual modalities. In this connection space, similar concepts lie close, while dissimilar concepts lie far apart, irrespective‘ of their modality. We discuss

four modality transformations: visual to text, text to visual, visual to visual and text to text. We introduce a novel attention mechanism to align multi-modal embeddings which are learned through a multi-modal metric loss function. The common vector space is shown to enable bidirectional generation of images and text. The learned common vector space is evaluated on multiple image-text datasets for cross-modal retrieval and zero-shot retrieval. The models are shown to advance the state-of-the-art on tasks that require joint processing of images and natural language.

Acknowledgements

There are many people I must thank for contributing to the wonderful years of my experience as a PhD student.

First and foremost, I am deeply indebted to Dr. Ray Ptucha, my teacher and advisor, for providing his valuable time and guidance throughout these years. His constant encouragement and support towards experimenting with new ideas really had me going. He was always willing to support and guide me on any research path. He was patient to go through the terrible paper drafts that I produced. I am greatly honored to have worked under him.

I would like to thank the members of my thesis advisory committee, Dr. Carl Salvaggio, Dr. Nathan Cahill and Dr. Leon Reznik, for providing their knowledge and time throughout this project. Thanks to Dr. David Messinger, Dr. Charles Bachmann and Dr. John Kerekes from CIS for their support throughout the graduate program.

My sincere thanks goes to the Center for Imaging Science and the Computer Engineering department at RIT for providing financial support throughout this degree. I also thank all the professors, staff, and all fellow students at the Imaging Science department at RIT, for the wonderful experience and knowledge I have received throughout the program.

During my PhD, I also squeezed in three summer internships at Xerox-PARC, Motorola and NVIDIA. All three were wonderful learning experiences that had a lot of impact on my research trajectory. From these internships, I would like to thank Bob Loce, Sean Kelly, John Poplett, Elaine Jin and Robin Jenkin.

My gratitude goes out to the RIT academic and IT staff including Sue Chan, Beth Lockwood, Joyce French, Kathryn Stefanik, Emilio Del Plato, Richard Flegal for their help.

This work would not have been possible without help and support of all the people around me, who were always kind and willing to give their time and valuable assistance towards the completion of this thesis. I would like to thank my close collaborators

who Ive had the distinct pleasure of working with and learning from. There are many others who I was fortunate to get to know and become friends with. From the MIL Lab: Sumanth Chennupati, Sai Nooka, Ram Oruganti, Suhas Pillai, Sourabh Kulhare, Naga Reddy, Ram Longman, Ameya Shringi, Felipe Such, Thang Nguyen, Dheeraj Peri, Rohan Dhamdhere, Sabarish Gopalakrishnan, Sidharth Makhija. Fellow PhD students: Chi Zhang, Miguel Dominguez, Srinivas Sridharan. From CIS: Bikash Basnet, Ritu Basnet, Viraj Adduru, Madhurima Bandyopadhyay, Saugata Sinha, Allison Gray, Yilong Liang, Daniel Simon. From UT Austin: Subhashini Venugopalan.

I thank Dr. Jan van Aardt for encouraging me to join the PhD program. I would also like to extend thanks to people who I was fortunate to interact with before joining the PhD program, especially Prof. A. K. Pachauri, Naval Kishore and Sushanta Jena from NDMA, New Delhi.

Lastly, my deepest gratitude goes to my family for their support throughout these years. They are, without a doubt, the key to my success.

This thesis is dedicated to my parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is easy for humans to accomplish a wide variety of tasks that involve complex scene understanding and visual recognition, tasks that involve communication in natural language and tasks that combine translation between the two modalities. For instance, a quick glance at an image is sufficient for humans to notice the immense amount of details about the visual scene and communicate that information using natural language. The creation and availability of large scale image and video datasets has seen tremendous growth with the machine learning revolution. Recent developments in convolutional and recurrent neural networks have led to unprecedented vision and language understanding. Steady advances in image classification [2, 3, 4, 5], object detection [6, 7, 8], semantic segmentation [9, 10, 11], and localized image description [12] led to some very elegant and powerful image captioning [13, 14, 15, 16] and video [17, 18, 19] captioning frameworks that have resulted in numerous deep networks capable of providing apt textual description of images and videos.

Applications in the consumer, medical, security, and military fields have seen tremendous growth in the past few years due to recent discoveries in deep learning. Hundreds of videos per minute are uploaded to *YouTube*, while the use of surveillance, automobile and body cameras are projected to increase dramatically. A few years ago one would have downloaded a repair manual to replace a broken tail light on a car, but

today, it is much more common to watch an instructional video. It is impossible for a human to effectively and efficiently utilize the voluminous amount of data without automated search, retrieval, summarization, and indexing methods. The efficacy of automated methods is contingent on their ability to understand the underlying content in the video. Towards this goal, there has been much research on video understanding. Tasks such as activity classification, video captioning, retrieval and object tracking have helped generate improved video analytics.

Early work on video captioning relied on extracting semantic content such as subject, verb, object, and associating corresponding visual elements [20, 21]. For instance, Thomason *et al.* [20] form a Factor Graph Model to obtain the probability for the semantic content and use a search based optimization to get the best combination to fit in a sentence template. Earlier works were also limited to activity or context specific videos with a small vocabulary of objects and activities. With availability of large video-sentence pair datasets with rich language information, recent studies [17, 13] have demonstrated the use of neural networks to directly model language conditioned on video.

Chapter 2 furthers the field of video understanding by introducing semantic video information in the captioning task. A robust captioning framework is introduced which can deal with both simple and complex videos. The main contributions in this chapter are four fold. Firstly, introduction of length agnostic Gaussian attention since existing soft attention models have an intrinsic limitation that all input buffers need to be of the same duration. This is because the attention vector is associated with a learnable, but fixed dimension weight matrix. For videos, this requires reducing longer videos or padding shorter videos. The proposed parametric Gaussian attention model removes this limitation by applying a continuous, rather than a discrete weight distribution. Secondly, using temporal features of a video to adaptively determine hierarchical transition points and allow a variable number of transitions from a granular (frame) level to a segment (clip) level. This forms an intelligent hierarchy for encoding a video that is

referred as the multistream hierarchical boundary model. Thirdly, proposing a temporal attention steering mechanism that uses frame level visual concepts to guide attention based on current video properties (activities, events, and detection of objects). Most existing attention models are guided by temporal features of the training data. Lastly, a real-time analysis of video captioning over varying video quality and frame rates is presented. A family of captioning frameworks are contrasted such that applications can make appropriate quality vs. speed trade-offs. The video fidelity and timing experiments suggest video captioning models are now suitable for automated surveillance systems in applications such as retail stores, amusement parks, power plants, and military installations.

Ease of use, instant sharing, and high image quality have resulted in abundant amounts video capture not only on social media outlets like Facebook and Youtube, but also personal devices including cell phones and computers. Several solutions are available to manage, organize, and search still images. Applying similar techniques to video works well for short snippets, but breaks down for videos over a few minutes long. In Chapter 3, the field of video captioning is advanced by leveraging several recent discoveries in the video summarization, video annotation, and text summarization fields, for summarizing very long videos.

The proposed method uniquely identifies interesting segments from long videos using image quality and consumer preference. Key frames are extracted from interesting segments whereby deep visual-captioning techniques generate visual and textual summaries. Captions from interesting segments are fed into extractive methods to generate paragraph summaries from the entire video. The paragraph summary is suitable for search and organization of videos, and the individual segment captions are suitable for efficient seeking to proper temporal offset in long videos. Because boundary cuts of interesting segments follow cinematography rules, the concatenation of segments forms a shorter summary of the long video. The method provides knobs to increase and/or decrease both the video and textual summary length to suit the application. While the

methods are evaluated on egocentric videos and TV episodes, similar techniques can also be used in commercial and government applications such as sports event summarization or surveillance, security, and reconnaissance.

An ambitious goal for machine learning and signal processing research is being able to represent different modalities of data that have the same meaning with a common latent representation. For example, words like "beach" and "ocean", a sentence describing a beach scene, a paragraph depicting waves crashing on a beach, and image and video representations of a beach all refer to a common concept. Concepts that are similar lie close together in this space while dissimilar concepts lie far apart. A sufficiently powerful model should be able to store similar concepts in a similar vector representation or produce any of these realizations from the same latent vector. One such application of image-text alignment has fueled the growth of new capabilities such as improved description of visual stimulus [22], advanced image and video search [23], and video summarization [24, 25]. Successfully mapping of visual and textual modalities in and out of this latent space would significantly impact the broad task of information retrieval.

Recent success in image captioning [13, 26, 14, 27] has shown that deep networks are capable of providing apt textual descriptions of visual data, thereby enabling a one-way path between modalities from image or video to text. In parallel, advances in conditioned image generation [28, 29, 30, 31] provide photo-realistic and diverse images from a text based prior. A common occurrence in the aforementioned domains is the presence of a latent vector representation that facilitates modality transition.

The task of generating an image has been made possible by Generative Adversarial Networks (GANs) [32] in which a generator is pitted against a discriminator which tries to classify the images as real or fake. However, such models are associated with complex learning mechanisms and demand large datasets. The adversarial loss used in GAN training is not indicative of the image quality and hence the generated images do not look visually appealing for challenging datasets like MS-COCO. In Chapter 4, we combine the networks used in such domains by merging the latent representations obtained during

transition. We demonstrate the efficacy of our model in within-domain and cross-domain transformations. The contributions in this chapter are three fold. Firstly, a latent representation based model is formulated that merges inputs across multiple modalities. Secondly, an n-gram based cost function is proposed that generalizes better to a text prior. Lastly, a sentence paraphrasing model capable of synthesizing similar sentences is trained and used to generate multiple sentences for conditioning image generation on generalized text. To evaluate the models, an inception score [33], proposed object detector based metric, and human evaluations are used. Results show that adding paraphrased sentences improves images quality across all three metrics. Along with quantitative evaluation, qualitative evaluation through text and image arithmetic in latent space is introduced. The results demonstrate mathematical properties exhibited by latent representations for certain objects.

In addition to the task of image and text generation, we also extend the common vector space model for cross-modal retrieval in Chapter 5. The contributions in this chapter are two fold. Firstly, we introduce a novel attention mechanism to align multi-modal embeddings which are learned through a multi-modal metric loss function. Secondly, we evaluate the learned common vector space on multiple image-text datasets for cross-modal retrieval and zero-shot retrieval. We extend the methodology to five different modalities- image, sentence, audio, video and three-dimension model and demonstrate multi-modal retrieval. We obtain state-of-the-art Mean Average Precision (mAP) scores for cross-modal and zero-shot retrieval to demonstrate the robustness of the trained common vector space.

The learnings from Chapters 2 and 3 helps in developing a system to attain vision-to-text transformations. Chapters 4 and 5 extends the ability for bidirectional transformations between the visual and text modalities. Using learning from these chapters, the research goals of this thesis are to understand the underlying content in video and represent different modalities of data that have the same meaning with a common latent representation.

Finally, in Chapter 6, we summarize the aforementioned works, identify the remaining challenges and discuss the path forward.

Motivations – The goal of connecting vision and language modalities can be motivated on a long-term scale of building intelligent machines, which would enable interaction between humans and computers in a natural and intuitive fashion. Developing such artificially intelligent agents require us to make large amounts of data, about our world available to computers. This data includes two main sources of knowledge- the physical world that is captured through sensors and includes scenes, objects and interactions; and the digital world of the Internet that contains vast amount of semantic information primarily in the form of images and text. Both these data sources complement each other. Therefore, vision and language are the two primary channels of knowledge through which information in the world can be accessed. It is very important that techniques are developed to relate information across these two channels rather than processing them independently.

The ambition to connect the vision and language modalities can also be motivated with short-term and practical application oriented arguments. Natural language offers appealing practical properties by representing nouns (objects, scenes, people), adjectives (attributes), verbs (actions) and nested constructs that assert relationships. Areas of computer vision such as classification of scene, attribute, action or objects are generalized by the task of natural language prediction from a visual input. The task of language prediction inherits all challenges faced by individual visual recognition tasks. Moreover, the end users of most computer vision systems are humans who are already familiar with natural language. Thus, using natural language as a bridge for learning vision problems enables natural and easy interactions between computers and humans. For instance, image search over the web using a query "person running in a park" would be more instinctive for a human compared to searching using intermediate stages for categories- "person", "run" and "park". The algorithms should be able to directly consume natural language as understood by humans, thus utilizing the rich encoding present in natural

language. Another direct application of a captioning system is describing or answering queries about a scene or movie to a visually impaired person.

Outline of Contributions – In this dissertation, we solve multi-modal vision and natural language tasks using a common latent representation through which models and algorithms communicate. For example, a captioning model should be able to take an image or video as input and describe the contents in natural language. Additionally, a model should be able to process an input natural language description and generate or identify the visual counterparts that depict the description. Overall, the goal is to connect the two modalities of vision and language through a common vector space such that translations between them is possible. In summary, in this dissertation we adopt the end-to-end learning paradigm and design neural network architectures for the tasks of image and video captioning, bi-directional image and text generation and cross-modal retrieval.

Chapter 2

Video Captioning

Before the advent of deep learning, automatic annotation of image and videos with natural language seemed years away. Subsequent research using attention mechanisms over spatial [27, 34], temporal [35, 36, 37] and attribute [38] domains localized focus to specific spatiotemporal locations to push the field further. While these attention mechanisms are one of the primary drivers for recent progress, our ability to understand how well temporal attention works on video is limited given that most datasets are comprised of short videos. For example, the average video duration of the MSVD [39] YouTube clips is 10.2 seconds and M-VAD [40] movie descriptions clips is 5.8 seconds.

Tran *et al.* [41] introduced VGG-like 3D convolutional nets for video feature extraction. Rather than learn a multiple C3D vectors, Pu *et al.* [42] introduced attention over intermediate convolution layers. Features from lower layers focus on fine-grained information while features at top of the CNN focus on global information. Rather than seek correlations between convolutional layers, our model extracts frame-wise visual concepts across the length of the video. This elegantly enables the model to correlate specific concepts such as woman, man, and skateboarding, with region-specific locations across the video.

As attention weights are learned parameters, and the number of parameters needs to be fixed at train time, attention models are constrained such that all samples have

equivalent dimensions. To learn and reproduce handwriting, Graves [43] introduced attention to arbitrary regions of the output by predicting parameters of a mixture model. To enable the attention mechanism to be independent of video duration, we present a Gaussian attention model which learns a continuous function and samples this function temporally into discrete regions.

The hierarchical abstraction afforded by deep neural nets enables the learning of activation maps of high and low spatial detail. Pan *et al.* [36] introduced a neural encoder for video captioning using a recurrent hierarchical partitioning structure to create a pyramid of abstract representations. However, the temporal transition between frames and clips is a fixed hyperparameter. We introduce an intelligent boundary learning scheme that helps to form an adaptive hierarchy for encoding a video. Our steered hierarchical Gaussian attention model uses an intuitive video2vec latent encoding. When applied to variable length videos in an adaptive hierarchical fashion, we can demonstrate state-of-the-art captioning results on the MSR-VTT [44], MSVD [39] and M-VAD [40] video captioning datasets.

This chapter is organized as follows: Section 2.1 reviews the relevant literature, Section 2.2 introduced the proposed Gaussian attention, the multistream hierarchy boundary model and attention steering approaches in detail, Section 2.3 overviews the complete video captioning framework and Section 2.4 discusses the experiments and the results.

2.1 Related Work

Success of deep learning in the still image domain has influenced research in the video understanding domain [45, 46]. Early work on video captioning relied on extracting semantic content such as subject, verb, object, and associating it with the visual elements [20, 21]. For instance, [20] used a Factor Graph Model to obtain the probability for the semantic content and then use a search based optimization to combine a subject, verb and object to fit a sentence template. Earlier works were limited to activity or context

specific videos with a small vocabulary of objects and activities. With availability of large video-sentence pair datasets with rich language information, recent studies [17, 13] have demonstrated use of neural networks to directly model language conditioned on video. Deep neural network architectures for video classification are now prevalent [47, 48].

Initial works that introduced Recurrent Neural Networks (RNNs) for video captioning used a mean pooled feature as the video representation [17]. An alternate approach uses an encoder-decoder [18] framework that encodes f frames, one at a time to the first layer of a two layer Long-Short-Term Memory (LSTM), where f can be of variable length. S2VT [19] encodes the entire video, then decodes one word at a time.

Attention mechanisms were initially proposed in [49] and used in video captioning context by [35]. They allow the focus of relevant temporal segments of a video conditioned on the text-generating recurrent network. Spatial attention over parts of an image was shown by [27]. They used the outputs of the last convolution layer to guide the word generation to look into specific regions of an image. They also presented a hard-attention mechanism equivalent to reinforcement learning with the reward for selecting the image region proportional to the target sentence. Semantic attention over word attributes was shown to enhance image captioning by [38]. Similarly, [50] and [51] included video attributes or tags to help generate improved captions. Dong *et al.* [50] used a tagging embedding to enrich the LSTM input and re-rank generated sentences by their relevance to a video. Rich object and motion video features have also been used in video captioning [52]. The attribute or tag selection is not trained along with the language model and it becomes challenging to obtain rich attributes or "concepts" for videos that can also categorize actions along with objects.

More recently, video captioning was extended to paragraph generation using independent recurrent networks at the word and sentence level [37]. Hierarchical recurrent networks have also been used to encode the video in an embedding before generating words [36]. However, the temporal transition to form the hierarchies are fixed. They

also applied the attention over multiple stages (local, regional and global) which increases the number of learnable parameters. All described methods were dependent on availability of large scale datasets with video-sentence paired data.

Knowledge transfer from independent language and image data for image captioning was demonstrated by [53]. Our work is loosely inspired by this study because we want to use sentence independent visual features to improve the generated captions. Our work is additionally inspired by the soft attention model for video captioning presented in [35]. We augment it by parameterizing the attention mechanism with a Gaussian distribution over the video length and then further guide the attention using independent temporal "concepts" of the video inspired by the word attributes from [38]. Gaussian attention filters are discussed in [54] but the application is limited to activity classification and their equally spaced attention filters limit the use of attention for word generation. Our model is length agnostic since each Gaussian learns normalized mean and sigma values from the distribution.

2.1.1 Attention Models

A simple way to encode video features is by averaging pixels or features across all frames in the video. Most commonly, features are the output of a frame passed into an ImageNet pre-trained CNN. Soft Attention (SA) uses a weighted combination of these frame-level features, where the weights are influenced by the word decoder. Soft attention was first used in the context of video captioning in [35]. They computed a frame relevance score $e_i^{(t)}$ for each frame i of video $v_1, v_2, ..., v_n$ at decoder time step t.

$$e_i^{(t)} = \mathsf{w}^\top tanh(W_a h_{t-1} + U_a v_i + b_a) \tag{2.1}$$

Where, h_{t-1} is the hidden state at the previous time step of the decoder, v_i is the frame feature vector representation of the i^{th} frame, and w, W_a, U_a, b_a are learned parameters. This can be interpreted as an alignment between the encoder and decoder sequence. It allows the video encoder to selectively emphasize relevant parts of the

video. As the frame relevance score is computed using fixed dimension weight matrices, it restricts the exact number of frames in the video. Moreover, given that the average length of videos is a few seconds in most datasets, it seems counter intuitive to have strong localized attention in such a short duration. As the attention is at a frame level, alignment of the most relevant video segment with the decoder sequence would yield more appropriate relevance scores.

2.2 Video Captioning

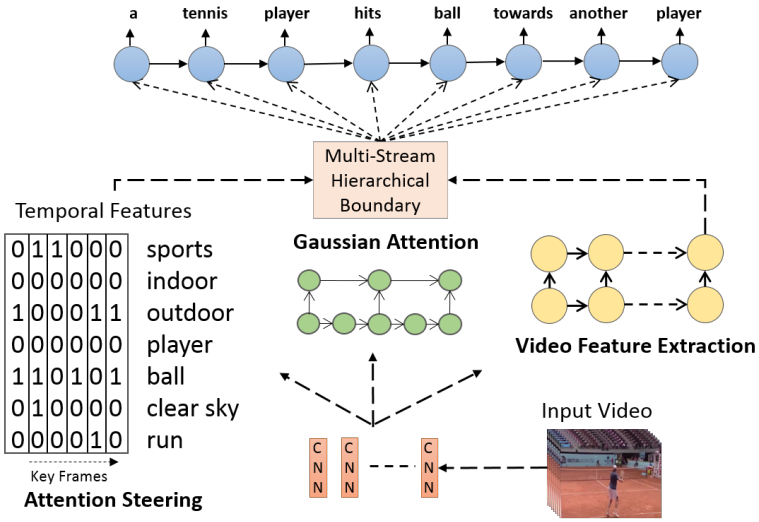


Figure 2.1: Overview of the Steered Gaussian Attention Model for video captioning. The attention filter is learned by hierarchical boundary model (center), temporal features (left), and a video summary (right).

This section describes the main components of the video captioning model- Gaussian attention, multi-stream hierarchical boundary model, attention steering and Video2Vec representation.

2.2.1 Gaussian Attention

We define the Gaussian Attention (GA) to remove restrictions with the generic soft attention mechanism. The relevance score that weighs the input sequence is modeled with a Gaussian distribution. At each time step, the decoder observes a filtered/weighted encoder sequence. GA weighs the input sequence based on the temporal location and the shape of the distribution modeled by the mean and standard deviation, respectively. We adapt the function to compute a continuous relevance score e^t across the entire input sequence $X = (x_1, x_2, ..., x_F)$ at decoder time step t as:

$$e^t = \sum_{k=1}^{N} \pi_k \mathcal{N}(X|\mu_k^t, \Sigma_k^t) \tag{2.2}$$

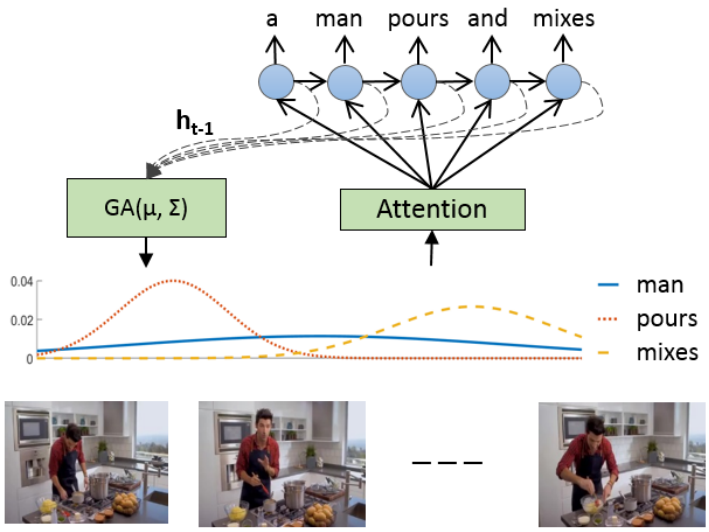


Figure 2.2: Illustration of the parameterized Gaussian attention model for steering the temporal alignment between the video and word sequence. The caption is generated using a recurrent neural network. For a video, the mean and standard deviation of the distribution is computed based on the outputs of the previous time steps (dotted lines). The curves depict change in the attention over the video based on the word generated in the caption generator.

where, each GA $\mathcal{N}(X|\mu_k^t, \Sigma_k^t)$ is a Gaussian distribution with its unique mean μ_k^t and

covariance matrix Σ_k^t at time t, N is the number of Gaussians and π_k is the mixing coefficient. The mixing coefficients are normalized to sum to one. The input features $X \in \mathbb{R}^{D \times F \times M}$, where D is the number of input modalities, F is the length of the each sequence, and M is the dimension of each feature. For example, if the two input modalities of spatial domain and temporal domain are used, we can learn a unique set of Gaussians for each modality by setting $D = 2$. By computing the mean and covariance of sufficient number of Gaussians, superposition can approximate any continuous function. Hence with correct parameters, a GA model can achieve the same function as soft attention. We choose to model independent Gaussians, and replace Σ_k^t with a scalar standard deviation, σ_k^t at each time t.

Computing the parameters allows the filter to temporally adapt to decoder decisions. With loss backpropagated at each time step, the mean value of the Gaussian learns to control focus on relevant locations of the sequence. Similarly, the standard deviation can learn to extract information from a longer or shorter segment. Thus, the GA formulation makes it adaptive both in terms of location and range. Resource utilization can be optimized as the decoder need not necessarily compute attention over the entire input sequence. The mean and standard deviation are computed as:

$$\mu^t = \wp(W_\mu h_{t-1} + U_\mu X + b_\mu) \tag{2.3}$$

$$\sigma^t = |W_\sigma h_{t-1} + U_\sigma X + b_\sigma| \tag{2.4}$$

where, W_μ, W_σ, U_μ, U_σ, b_μ, b_σ are learned weights. We use the activation $\wp(s) = |s|/(|s|+c)$ for the mean values to scale to range [0,1] as the input sequence is normalized temporally, where c is a hyper-parameter. The normalization allows the model to compute attention over sequences of varying length. It also reduces the number of learnable weights from $\mathbb{R}^{h \times h}$ to $\mathbb{R}^{h \times N}$, where h is hidden dimension size of decoder and $N \ll h$. The activation for the standard deviation σ is different since we do not need to

scale these values other than constraining them to be positive. Similar to soft attention, the attention weights α_i^t at time t for input X are obtained by normalizing the relevance scores. The input to the decoder is a weighted sum of the input X using the attention at time t.

$$\alpha_i^t = \frac{exp(e_i^t)}{\sum_{j=1}^{F} exp(e_j^t)} \tag{2.5}$$

$$\Phi_t(X) = \sum_{i=1}^{F} \alpha_i^t x_i \tag{2.6}$$

Modeling the attention filter with a parametric distribution allows the decoder to view inputs with varying duration and hence it is better at exploiting the temporal structure of an input sequence. The parametric attention has the capability to sense the complete encoder sequence if required. This is important in a translation like task where the generated word may hold relevance throughout the video. For example, after the word *man* in Figure 2.2, the model learns to expand the attention to allow the caption generator to view the entire input as the associated visual feature of *man* appears in the entire video.

2.2.2 Multistream Hierarchical Boundary Model

In hierarchical models, the output of local features from the first layer are input to the second layer in a fixed stride style over short video chunks [36]. This is demonstrated by the local and hierarchical features as the bottom two layers in Figure 2.3. The fixed stride may mix up several shots with no related features. [55] proposed a boundary detection unit to learn individual segments, resetting the prior RNN state after each segment. This method has a drawback in cases of videos with no natural boundaries- in such cases it could not leverage the intrinsic temporal dependencies in the video stream.

To deal with clips with different structures, we propose a Multistream Hierarchical Boundary (MHB) model which can take full advantage of both the hierarchical and

boundary architectures. The MHB model consists of multiple stages, with the encoder stage transforming video frames to a vector representation and the caption decoder stage transforming those vectors into arbitrary length sentences.

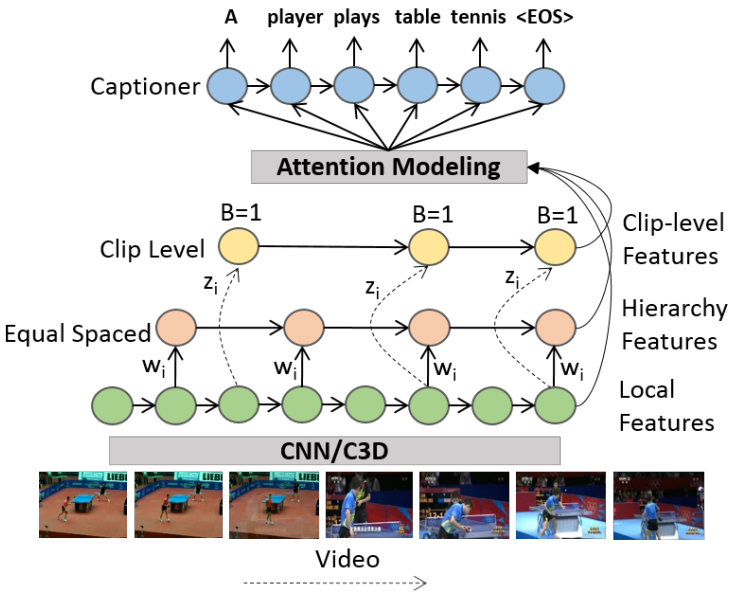


Figure 2.3: Overview of the Multistream Hierarchical Boundary Model for video captioning. The clip-level features adapt with each video and are learned non-equally spaced and the hierarchy features are equally spaced features.

Referring to Figure 2.3, the encoding stage takes in a given video stream, whereby the first layer takes in local features $(x_1, x_2, ..., x_n)$, and outputs two sequence vectors: 1) equal spaced $(w_1, w_2, ..., w_p)$; and 2) clip level $(z_1, z_2, ..., z_q)$. The equal spaced output layer gets p outputs from first layer with $p = n/k$ (n is number of input features and k is designed stride value). The clip level output layer utilizes information on shot boundaries guided by a learned vector based on the cosine distance:

$$z_i = y_i.(\Delta(i, j).W_{yd} + b_{yd}) \tag{2.7}$$

where W_{yd} and b_{yd} are learned weights and bias, y_i is output at each time step of first layer. As illustrated in Figure 2.3, the video is encoded through a combination of equally

spaced and clip level feature representations. The fusion of local (frame) level, hierarchy (equally spaced) and clip (detected boundaries) level is input to the caption decoder. At each time step, the model adapts the boundary weights to extract information from the relevant segments of the video thus being extremely efficient in encoding complex video sequences. We incorporate Gaussian attention at the equally spaced and clip level hierarchies.

Shot Boundary Detection– Features extracted from CNN models have proven to be useful in cut-transition boundary detection between two shots in a video stream [56]. Given α_i and α_j are two CNN feature vectors of two consecutive frames, the cosine distance $\Delta(i, j)$ between them can be calculated as follows:

$$\Delta(i,j) = \cos(\alpha_i, \alpha_j) = \frac{\alpha_i.\alpha_j}{\|(\alpha_i).\|(\alpha_j)} \tag{2.8}$$

where, $\Delta(i, j) \in [0, 1]$. Higher values indicate higher probability of a boundary cut. For example, one could experimentally determine a threshold ζ where a boundary exists when $\Delta(i, j) > \zeta$. Unlike Euclidean distance, the cosine distance needs no additional normalization steps. Xu *et al.* [56] determined this distance is effective in the cut-transition detection task. Our results concur and we employ it to detect the boundary to facilitate the soft hierarchy layer.

Figure 2.4 demonstrates this concept in an example video. The cosine similiarities of compared frames are tracked until the threshold is passed, signifying a change in scene. The threshold is determined as a hyper-paramter.

2.2.3 Attention Steering

Traditional attention models are associated with a set of weight matrices that are learned during training. During test time, the weight matrices guide the attention and hence limit the attention mechanisms by prior temporal statistics. We introduce temporal attention steering that guides the attention based on the visual features of a test video

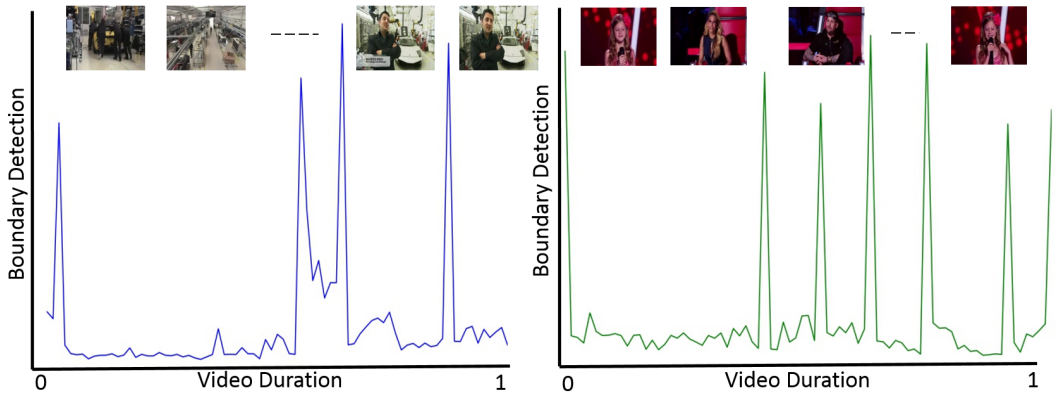


Figure 2.4: Example boundary attention vector where the "peaks" indicate video boundaries in $[0-1]$ normalized video.

(Attention Steering module on left side of Figure 2.1). The temporal features across the video are normalized over all frames. The resulting matrix is a temporal map that translates feature relevance to frame relevance. At each LSTM time step, the model computes an updated frame relevance vector. For example, if the network computes that "apple" is an important feature for the next word prediction, the relevance factor of the feature "apple" will be higher. The temporal feature map in Figure 2.5 would then translate the relevance factor of "apple" to the center/end of the video. This provides a way to steer the attention without increasing the number of inputs to the system.

We investigate the use of word label embeddings of objects present in video frames as temporal visual features. We use an ImageNet classifier trained on 4k classes [57] represented using a GloVe [58] word embedding. This embedding was built on 400,000 vocabulary entries pre-trained on a 6 Billion word corpus from *Wikipedia* and *Gigaword*. Representing a large number of objects is important for "in-the-wild" videos. A bottom-up grouping strategy [57] is applied to the categories to deal with the problems of over-specific classes. In reality, a sentence is described by both the objects and the whole scene as the context. Distinguishing individual objects from others in a scene, especially when there are multiple objects of different categories, can be highly challenging. Hence,

EdgeBox is used [59] to obtain proposal bounding box regions within each frame of a video. For the top 95% of all bounding boxes, we compute GloVe word embedding of the ImageNet 4K CNN classes. The GloVe word embeddings of bounding box class labels are mean pooled to obtain a frame-level representation. We discover that the mean pooled class label embedding is rich in semantic information and is closer to the words in the ground truth sentence. Moreover, use of word embedding reduces the feature dimension from 4K to 300. This design choice reduces number of parameters to be learned substantially. As a complementary or alternative approach to temporal word embeddings, one could use frame CNN features directly.

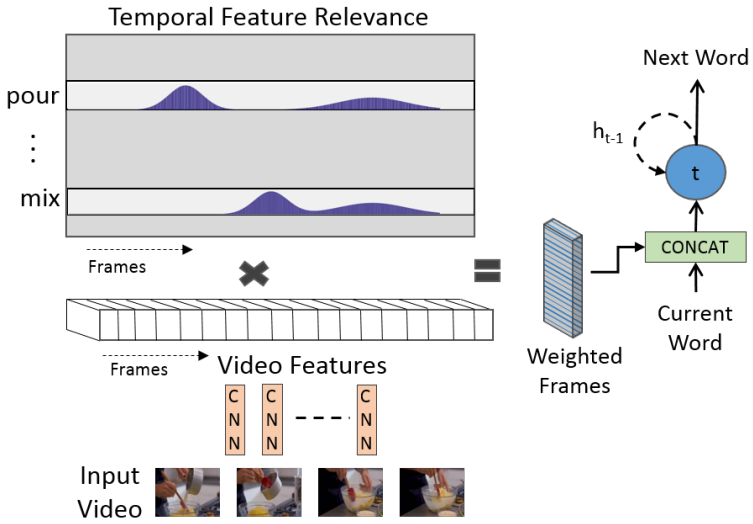


Figure 2.5: Attention steering using normalized temporal feature relevance. Frame level features are weighted based on the relevance map and assists in guiding attention to video regions. W_t and W_{t-1} are words at times t and $t-1$, h_{t-1} is RNN hidden state.

Video2Vec Representation– In addition to the steering mechanism, an embedded vector representation of the entire video is input into the captioning model (right input in Figure 2.1). To learn powerful action and motion representations, we use a recent activity classification dataset- ActivityNet [60], on human activity understanding that covers a wide range of complex daily activities. It is comprised of 849 video hours in

over 200 activity classes. As these videos were collected from online video sharing sites they are excellent to transfer learned features for MSVD and MSR-VTT datasets which are also based on Youtube videos. The labeled videos are used to train a standard video-based activity classifier. We utilize two independent models with RGB (3- color channels) and Optical Flow (OF) inputs. Features before the loss layer are used as *Video2Vec-Activity* representation. We fine-tune the last fully-connected layer during caption generation.

2.3 Video Captioning Framework

Hierarchy with Gaussian Attention– The proposed MHB technique efficiently captures temporal dependencies in videos. Hence, we integrate it with our GA model and term it as Hierarchy with Gaussian Attention (HGA). The hierarchy of recurrent layers adds more non-linearity to the GA model. The hidden state of LSTM in layer $l-1$ at the last time step is the input to layer l. This ensures easy back-propagation of loss compared with a simple layer stacking by reducing the number of steps the loss back-propagates. The first layer learns local temporal dynamics within short clips and the second layer learns the difference between these short clip sequences. The output at the last time step of the second layer is a vector representation for the entire video.

The video captioning framework has three main components – Attention Steering, Video2Vec encoder and Hierarchical Gaussian attention based sentence generator as shown in Figure 2.1 (left, right and center). The sentence generation engine takes input from all three to generate word sequences. Recurrent Neural Networks (RNN) are a natural choice for generating sequences such as natural language sentences. However, RNNs suffer from vanishing and exploding gradient problems when learning long sequences. To solve this, we use the LSTM variant of RNNs to learn sentence generation as it is known to learn sequences with both short and long temporal dependencies [13].

The model is trained using stochastic gradient descent by learning parameters θ

for the sentence $w_1, w_2..., w_\tau$. The word-based loss for a video is $loss_{caption}$. The log-likelihood is optimized by minimizing the loss for video V with word embeddings V_c and Video2Vec embedding S_v.

$$\theta^* = \max_\theta \sum_{t=0}^{\tau+1} log(w_t|V, S_v, V_c, w_{t-1}; \theta) \tag{2.9}$$

where w_0 and $w_{\tau+1}$ are special tokens for start and end of sentence. During testing, the model is input with the token for beginning of sentence and it generates words until the end of sentence token is generated.

Word Feature Loss – Inspired by the work in [61] on multi-modal embedding between text and visual inputs, we compute the cosine similarity between the mean pooled video level word embedding (V_c) and Gaussian attention weighted video vector (Φ_V). This similarity measure is added to the caption generation loss for the entire video using (2.8), replacing α_i and α_j with V_c, and Φ_V.

2.4 Results and Discussion

2.4.1 Training Details

Each video frame is passed through the 152-layer ResNet CNN model [3] pre-trained on the ImageNet dataset [62], where the $[1 \times 2048]$ vector from the last pooling layer (*pool5*) is used as the visual feature vector. In our HGA model, each batch of 12 frames input into the first hierarchical layer yield a single input to the second hierarchy layer.

We preprocess all words in captions with the PTBTokenizer in the Stanford CoreNLP tools [63]. This toolkit converts all text to lower case, removes punctuation, and tokenizes the sentences. We use captions only from the training and validation set to generate the vocabulary. Words start as one-hot encoding. For MSR-VTT video categories, we use 300-dimension GloVe embedding [58] to obtain word vector representations.

The architecture is implemented in *TensorFlow* [64]. During training, ADAM optimization [65] is used to minimize the negative log likelihood loss. The learning rate

is 2×10^{-4} and we use decay parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The dimension of the LSTM hidden vectors is 1024 for HGA and 384 for the sentence generation layer. We employ a Dropout [66] probability of 0.5 on the output of all LSTM layers. The mini-batch size is 100 videos and all models are trained for 40 epochs. Hyperparameters are evaluated on the validation set.

Beam Search − The LSTM generates a single word at each time step. Instead of a greedy search for the most probable word, we employ beam search at test time to yield a wider variety of sentences. A beam width of k produces a list of k top words at time step t for each of k partial sentences. The top k most probable sentences from these k^2 candidates are pushed forward to the next time step and the remainder are dropped. Empirically, a beam width of 10 with MSVD and 20 with MSR-VTT performs best. We suspect the large vocabulary size for MSR-VTT required higher beam width. This is in agreement with [50].

2.4.2 Datasets

We train and evaluate our models on the Microsoft Video Description Dataset (MSVD) [39], the newly released Microsoft Research - Video to Text (MSR-VTT) [44] and the movie description datset M-VAD [40]. Standard train, validation and test splits were used for all datasets. Table 4.1 summarizes the high-level properties of each dataset.

Table 2.1: Video-sentence pair dataset statistics.

	MSVD	MSR-VTT	M-VAD
#sentences	80,827	200,000	54,997
#sent. per video	∼42	20	∼1-2
vocab. size	9,729	24,282	16,307
avg. length	10.2s	14.8s	5.8s
#train video	1,200	6,513	36,921
#val. video	100	497	4,651
#test video	670	2,990	4,951

2.4.3 Evaluation Metrics

Quantitative evaluation was performed using the Microsoft COCO caption evaluation tool [67] to make our results directly comparable with other studies. This tool computes standard captioning metrics: BLEU [68], METEOR [69] CIDEr [70] and ROUGE [71] to score a predicted sentence against all ground truth sentences.

BLEU is a metric for precision of word n-grams between predicted and ground truth sentences. ROUGE takes into account sentence level structure similarity naturally and identifies the longest co-occurring sequence in n-grams automatically. METEOR was designed to fix some of the problems found in the more popular BLEU metric, and also produce good correlation with human judgment at the sentence or segment level. It has several features not found in other metrics, such as stemming and synonymy matching, along with the standard exact word matching. CIDEr computes the average cosine similarity between n-grams found in the generated caption and those found in reference sentences, weighting them using TF-IDF. METEOR is more semantically preferred than BLEU and ROUGE.

Typically, the generated sentence correlates well with a human judgment when the metrics are high as they measure the overall sentence meaning and fluency. However, the reliability of these metrics are ultimately subject to the mutual agreement between the visual input and the ground truth sentences. A model that learns from nonsensical sentences may accurately recognize patterns and achieve high scores, but will not be useful in practice. We report all scores as percentages.

2.4.4 Performance on MSVD

Table 2.3 reports current captioning results (top third) vs. variations on the number of Gaussians (middle third) vs. variations on MHB models (bottom third) on the MSVD dataset. These will be discussed next.

Gaussian Attention on MSVD

With respect to Table 2.3, middle third, we evaluate our Gaussian Attention module. Our baseline model (Baseline GA-5) is a Gaussian attention with five Gaussians. The addition of hierarchical modeling (+HGA) improves all scores. Overall, the HGA model learns bigger-picture motion features that compliment the more locally-focused Gaussian attention.

As recommended in [37], we test a variant with BLEU-4 score included in the caption loss (BLEU reg). The BLEU score is computed on the validation set and regularized with the loss after each mini-batch. Though it significantly improves BLEU-4 score, other scores are not much affected and we notice that sentence fluency degrades as well.

The addition of Video2Vec-Activity (+RGB,OF) further improves METEOR scores. These features give us extra motion understanding, as well as an understanding of the action concepts in Activitynet. The highest METEOR score that we achieve using GA is 33.1% which matches the state-of-the-art.

The advantage of learning multiple Gaussians is that more complex functions can be represented. This also allows, for example, multimodal distributions or distributions with a more discriminative shape than two parameters will allow. We evaluate the quality of the captions as the number of learned Gaussians is increased. Results are reported in Table 2.2. More Gaussians increase METEOR and BLEU scores across the board. We would have increased the number of Gaussians even further but ran into exploding gradients beyond five Gaussians.

Table 2.2: Performance evaluation with number of Gaussian filters for attention on the MSVD test set.

# Gaussians	METEOR	B-1	B-2	B-3	B-4
1	30.7	76.3	62.3	50.3	39.0
3	31.2	77.6	64.1	53.0	42.1
5	**31.5**	**80.4**	**66.6**	**54.5**	**42.8**

Figure 2.6 shows words from generated sentences along with a single temporal Gaus-

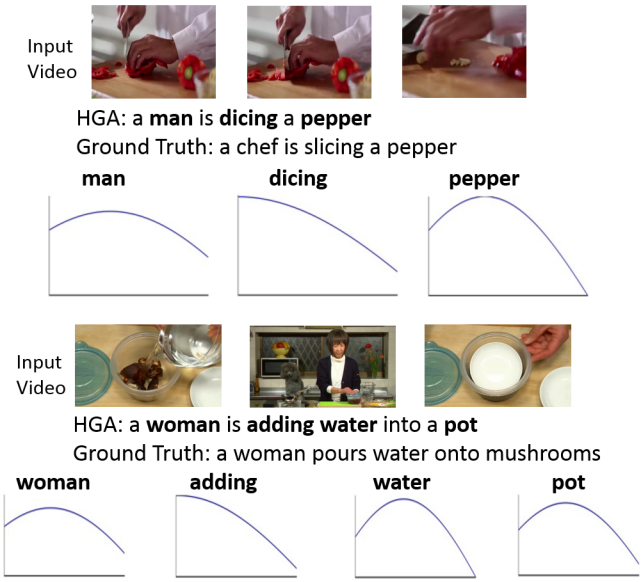


Figure 2.6: Gaussian attention visualization for sample videos from MSVD. Distribution focuses on relevant video segment based on key words (bold) in the sentence. For the word "adding", relevant activity is in the starting of video, hence the mean of the distribution is close to 0. X-axis ranges from $0-1$ normalized temporal video location and Y-axis is normalized attention weight α_i^t.

sian attention distributions generated on sample test MSVD videos. The distribution shows the adaptable nature of Gaussian attention. Even though the videos are short, at certain times the model needs to attend to different parts of the video. We anticipate that with longer and more complex videos, a higher number of Gaussians would be required.

Multistream Hierarchal Boundaries Model Experiments

We turn our focus to evaluating the Multistream Hierarchal Boundaries (MHB) Model. The bottom of Table 2.3 shows that our results on MSVD achieve the state-of-the-art METEOR score. MHB is a complex architecture combining many ideas, so an ablation study is necessary to evaluate the contribution of each idea. We compare the baseline architecture (MHB) with variants that remove individual features. MHB w/o

Table 2.3: MSVD caption evaluation results on the held out test set. All scores are reported in percentage.

Method	METEOR	BLEU-4	CIDEr	ROUGE-L
S2VT [19]	29.8	-	-	-
SA [35]	29.6	41.9	51.67	-
p-RNN [37]	32.6	**49.9**	65.8	-
HRNE Att [36]	**33.1**	43.8	-	-
Baseline GA-5	31.5	42.8	65.8	67.9
GA-5+BLEU reg	31.3	43.8	64.9	68.4
GA-5+HGA	32.8	43.9	**74.7**	**69.3**
GA-5+HGA+RGB,OF	33.1	43.0	71.1	68.8
MHB w/o GA	30.2	39.8	62.0	65.4
MHB w/o Bdr	32.5	42.3	68.6	68.2
MHB w/ LSTM	32.9	42.3	70.4	68.6
MHB	**33.2**	43.0	71.1	68.7

GA removes Gaussian Attention from the boundary layer. This means that the only attention in the model is the fixed-length soft attention in the equally-spaced layer. MHB w/o Bdr removes the Boundary layer entirely, leaving only the equally-spaced layer. MHB w/ LSTM replaces the Recurrent Highway Network cells with traditional LSTM cells. Omitting the Boundary Layer or Recurrent Highway Network cells incur small reductions in performance across all common captioning metrics. Omitting the Gaussian Attention alone causes the biggest reduction in performance. This is interesting because the Omitting the Boundary layer entirely also causes Gaussian Attention to be omitted, but is less damaging. This suggests that the Boundary Layer is only worthwhile if a flexible attention mechanism is applied to it.

2.4.5 Performance on MSR-VTT

Caption evaluation scores for our models on the MSR-VTT dataset are reported in Tables 2.4. All our models are trained end-to-end. A single layer GA performs better than the mean pooled video frame input features. The HGA model adds hierarchy features to a single layer GA model and hence is better at learning temporal depen-

dencies. The significance of Gaussian attention is shown by comparison of HGA with and without attention. This has better performance than a weighted average through an attention mechanism. To study the importance of temporal steering (STE) and Video2Vec-Activity (RGB and OF) features, we also input these as features to the captioning model. All of these inputs have positive impacts on the evaluation metrics. The addition of activity features show clear improvement over the baseline HGA. The OF features yield slightly improved scores over RGB. This indicates that motion/activity features from the ActivityNet dataset generalize well to other datasets.

Across all features, we observe that the scores did not change significantly when trained without word features loss (as in Section 2.3). However, it helped the model to converge faster. While generating the vocabulary from the training captions, we note that out of total 24,282 words, 10,155 words appear just once and 3,211 words twice. From the vocabulary, 4,716 words were not part of the GloVe 400K dictionary. Such issues add to challenges of the language model. Similar trends appear in other datasets as well.

Table 2.4: MSR-VTT results on the held out test set. We compare with recent entries in the MSR Video to Language Challenge.

Method		METEOR	BLEU-4	CIDEr	ROUGE-L
Dong *et al.* [50]		26.9	39.3	45.9	58.3
Multimodal (only visual input) [72]		27.0	38.3	41.8	59.7
Shetty and Laaksonen [52]		27.7	**41.1**	**46.4**	59.6
Mean pool		25.4	34.1	35.8	57.7
Ours	MHB	27.3	37.8	42.6	58.8
	Only GA 1-layer	25.6	34.6	37.4	57.4
	HGA (w/o att)	26.6	36.0	38.9	58.4
	HGA	27.4	38.8	43.4	59.1
	+ STE	27.6	37.9	43.4	59.2
	+RGB	27.6	38.6	42.8	58.9
	+OF	27.7	39.0	43.8	59.6
	+RGB, OF	27.7	39.2	43.5	59.2
	+RGB, OF, CAT	**28.2**	40.5	45.3	**60.4**

Fusion based models – Although the METEOR score does not improve with a

combination of RGB and optical flow features, all other metrics show improvement. It also indicates that either of the features are sufficient to capture the activity information. We also use the GloVe embedded video category label (CAT) available for all videos. The combined model is trained by concatenating the features before input to the LSTM. We note that the categories are the ground truth labels that are part of the original dataset and hence are better than any features generalized from an another dataset.

Gaussian Attention in Different HGA Layers Experiments were run on the HGA model to compare soft (SA) and Gaussian attentions (GA). The HGA-only model can be interpreted as a three layer LSTM with the first two hierarchical layers as the video encoder and the last layer as the sentence generator or word decoder. We replace soft attention with GA at multiple layer combinations. Results are reported in Table 2.5. Adding GA at more layers seem to help focus on relevant inputs and features. Attention on the middle HGA layers can be viewed as the weighted sum of the encoded outputs of video clips input to the first layer. Attention is most important at the word decoder (layer 3) as it not only finds relevant segments in the video but also relevant HGA encoded features based on generated words.

Table 2.5: Comparing Gaussian attention at different layers for MSR-VTT test set. Adding GA show clear improvement over SA and attention is most important at the word generation layer. B-1 to B-4 are n-gram BLEU scores.

Layer replacing SA with GA	METEOR	B-1	B-2	B-3	B-4
None	26.7	77.7	62.6	48.4	36.1
3	27.3	78.9	63.6	49.8	38.1
3,2	**27.4**	79.3	64.5	50.8	**38.8**
3,2,1 (HGA)	**27.4**	**79.7**	**64.8**	**51.1**	**38.8**

2.4.6 Performance on Movie Description Dataset

We present results of the HGA model on the M-VAD movie description dataset. This is a very challenging dataset as the videos are not specific activities but are movie scenes with complex sentences. We obtain a METEOR score of 6.9%, which is an improvement

over the HRNE (6.8%) [36] and S2VT (6.7%) [19] models. The BLEU scores are 17.3%, 6.0%, 2.7%, 1.0% for 1,2,3,4 − grams, respectively. MHB results in a METEOR score of 6.6%. This model is disadvantaged due to poor alignment of the ground truth captions with their respective video frames, which sometimes results in confusing cut-transition boundaries.

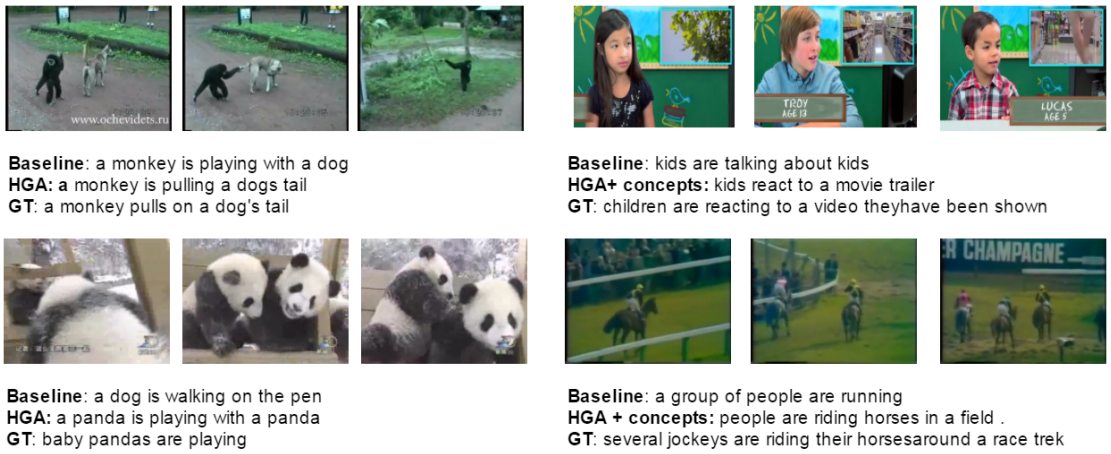


Figure 2.7: Example videos and corresponding captions from the MSVD (left) and MSRVTT (right) datasets. For each video, three random frames are shown. Baseline is GA model with five Gaussian filters. HGA is our model and GT is a sample ground truth caption.

2.4.7 Timing Comparison

For a video captioning tool to be useful in the field—say, for a security system, the captioner needs to be able to both produce high-quality captions and be compute friendly. In this section we conduct a set of experiments to benchmark the testing time and scores in selected models with various setups. Our machine specifications: Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz, RAM 128GB, GPU Tesla M40 with 24GB memory. We supply batches of 100 videos into our network and time how long it takes to produce captions from the entire 100-video batch. We vary the number of frames per video and measure the effect on processing time and quality. The network is retrained to be

optimized for a specific number of frames.

Table 2.6: Run(test) time for various models by varying number of frames per video for MSVD dataset.

Model	Running time (sec.)				
# frames per video	10	30	50	100	120
SA	3.46	3.85	4.56	5.21	5.47
GA	3.97	4.69	5.3	6.86	7.43
HRNE	-	-	46.98	48.70	49.45
MHB	-	-	50.81	53.6	54.65
HGA	-	-	48.61	50.12	51.4

Our speed comparison in Table 2.6 reflects that our straightforward GA model achieves comparable speed against SA models [35]. Both of these models can compute batches of captions multiple times a minute. Our more complex MHB and HGA architectures can supply a batch of captions slightly faster than once a minute. We vary the number of frames per video between 10 and 120. Since the average video in MSVD is 10.2 seconds, this translates to between 4% and 49% of the video. It has a negligible effect on total runtime.

Table 2.7 shows the effect of this reduction in frames on caption quality. The number of frames appears to have little affect on the final scores for all of the architectures evaluated. When taken together with Table 2.6, we start to get a picture of the practicality of these deep learning captioning systems in the field. The SA system of [35] and our GA system can produce high quality captions for 100 video feeds multiple times per minute. Our more complex MHB and HGA offer a bump in caption quality, but at slower speeds. These video feeds can operate at a duty cycle of 4% without a significant drop in captioning quality, providing an opportunity for significant power savings for the camera systems.

Table 2.7: Meteor score for various models by varying number of frames per video for MSVD dataset.

Model	# frames per video				
	10	30	50	100	120
SA	31.1	31.2	30.9	31.3	31.5
GA	31.2	31.2	31.3	30.9	31.5
HRNE	-	-	32	31.8	32.4
MHB	-	-	32.7	32.4	32.6
HGA	-	-	32.5	32.7	32.8

2.4.8 Compression Comparison

A surveillance system would likely face bandwidth challenges in trying to transmit 100 high-quality videos to a single PC. It would be convenient if these videos could undergo aggressive lossy compression and still be effectively captioned. We transcode video from the test set with *ffpmeg* [73], changing only the Constant Rate Factor (CRF). This is a quantization parameter where 0 induces no loss and 51 induces the most aggressive quantization possible. In our case, CRF=16 was the original setting. Increasing the CRF by 8 roughly equates to halving the bit rate. The number of frames in each experiment was chosen based on the best performing architecture in Table 2.7. We show in Table 2.8 that all of the architectures under evaluation can survive an increase of CRF of 20 (roughly dividing the original video's bit rate by 10) with minimal impact on quality.

Table 2.8: Meteor score for various models by varying video fidelity for MSVD dataset.

Model	CRF				
	16 (orig.)	24	36	48	64
SA	31.5	31.3	30.7	26.2	25.3
GA	31.5	31.4	30.9	25.6	24.0
HRNE	32.4	32.2	31.2	26.2	24.5
MHB	32.7	32.1	31.1	26.5	25.2
HGA	32.8	32.3	31.3	26.5	24.6

Chapter 3

Summarizing Long Videos

While computer vision techniques have significantly helped in organizing and searching still image data, these methods do not scale directly to general purpose videos, and are often computationally inefficient. Videos that are tens of minutes to several hours long remain a major technical challenge. To mitigate such problems, we propose techniques that leverage recent advances in video summarization [74, 75, 76, 24, 25], video annotation [77, 35, 78], and text summarization [79, 80], to summarize hour long videos to a substantially short visual and textual summary.

The novel contributions in this chapter include: 1) The ability to split a video into superframe segments, ranking each segment by image quality, cinematography rules, and consumer preference; 2) Advancing the field of video annotation by combining recent deep learning discoveries in image classification, recurrent neural networks, and transfer learning; 3) Adopting textual summarization methods to produce human readable summaries of video; and 4) providing knobs such that both the video and textual summary can be of variable length.

This chapter is organized as follows: Section 3.1 lists the related work, Section 3.2 describes the proposed methodology including the superframe segmentation framework and key frame selection and Section 3.3 discusses the results.

3.1 Related Work

Video summarization research has been largely driven by parallel advancements in video processing methods, intelligent selection of video frames, and start-of-the-art text summarization tools. [81] generates story driven summaries from long unedited egocentric videos. They start with a *static-transit* procedure to extract subshots from a longer egocentric video and extract entities that appear in each subshot to maximize a order of k selected subshots while preserving influence over time and individual important events. In contrast, [76] works with any kind of video (static, egocentric or moving), generates superframe cuts based on motion and further estimates interestingness of each superframe based on attention, aesthetic quality, landmark, person and objects. [82] uses video titles to find most important video segments. [24] explores a *nonparametric* supervised learning approach for summarization and transfers summary structure to novel input videos. Determinantal Point Processes which balances importance and diversity over a video using a distribution over the ground set has also often been used in video summary methods [83, 84, 24].

Using key frames to identify important or interesting regions of video has proven to be a valuable first step in video summarization. For example, [75] used temporal motion to define a visual attention score. Similarly, [74] utilized spatial saliency at the frame level. [76] introduced cinematographic rules which pull segment boundaries to locations with minimum motion. [85] favored frames with higher contrast and sharpness, [86] favored more colorful frames, [87] studied people and object content, while [88] studied the role facial content plays in image preference. [87] further tracked objects across a long video to discover story content.

Large supervised datasets along with advances in recurrent deep networks have enabled realistic description of still images with natural language text [13, 14, 26, 89]. The extension of this to video can be done by pooling over frames [77] or utilizing a fixed number of frames [35]. [35] uses a temporal attention mechanism to understand the global temporal structure of video, in addition they also use appearance and action fea-

tures through a 3-D Convolutional Neural Network (CNN) which encode local temporal structure. Most recently, [78] described a technique, S2VT, to learn a representation of a variable sequence of frames which are decoded into natural text. Recently, [37] demonstrated a hierarchical recurrent neural network to generate paragraph summaries from relatively long videos. These videos were still limited to a few minutes long. We use a variation of the S2VT captioning approach in our work.

Automatic text summarization systems are designed to take a single article, a cluster of news articles, or an email thread as input, and produce a concise and fluent summary of the most important information. Seminal summarization research by Luhn [90] and Edmundson [91] have spawned newer methods such as LexRank [92], SumBasic , and KL-Sum [93]. A good review of these techniques can be found in [94, 79]. The latest research on single document summarization has utilized both dependency based discourse tree trimming [95] as well as compression and anaphoricity constraints [80].

Given descriptive captions at key frame locations, we explore extractive methods for summarization. Extractive methods analyze a collection of input text to be summarized, typically sentences. These sentences are selected to be included in the summary using various measurements of sentence importance or centrality. Early seminal summarization research by Luhn [90] used word frequency metrics to rank sentences for inclusion in summaries, while Edmundson [91] expanded this approach to include heuristics based on word position in a sentence, sentence position in a document, and the presence of nearby key phrases. More recent extensions of the word frequency models, including SumBasic [96] and KL-Sum [93], typically incorporate more sophisticated methods of combining measures of word frequency at the sentence level and using these composite measures to rank candidate sentences. Other approaches, such as LexRank [92] and TextRank [97] focus on centroid-based methods of sentence selection, in which random walks on graphs of words and sentences are used to measure the centrality of those sentences to the text being summarized. A good review of these techniques and others can be found in [94][79]. The latest research on single document summarization has

utilized both dependency based discourse tree trimming [95] as well as compression and anaphoricity constraints [80].

3.2 Summarizing Long Videos

Videos of several hours long are frame averaged, then passed into a superframe segmentation algorithm. Each superframe segment is evaluated based in certain measures like- boundary motion, superframe motion, contrast, saturation, sharpness, and facial content. The top interesting superframe segments are then passed into an annotation module. The annotation module receives temporal segments, centered on each of the top superframe segments, and generates captions. After simple parsing, captions are then passed into the summarization tool, which outputs a single summary paragraph per video. The input consists of a single several hour long video. The output consists of a condensed video and a natural language summary paragraph.

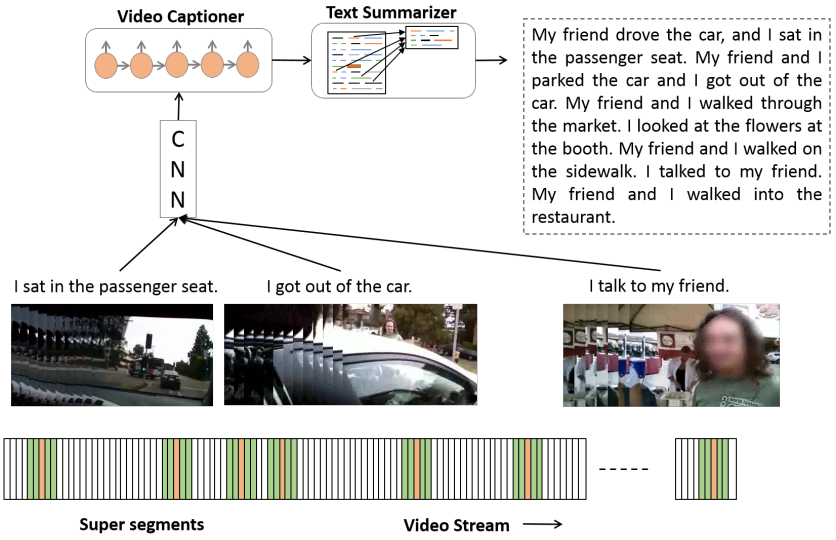


Figure 3.1: Overview of video summarization. Interesting regions identify key superframe segments. Each key segment is annotated. All annotations are fed into a text summarization module.

The proposed method uniquely identifies interesting segments from long videos us-

ing image quality and consumer preference. Key frames are extracted from interesting segments whereby deep visual-captioning techniques generate visual and textual summaries. Captions from interesting segments are fed into extractive methods to generate paragraph summaries from the entire video. The paragraph summary is suitable for search and organization of videos, and the individual segment captions are suitable for efficient seeking to proper temporal offset in long videos. Because boundary cuts of interesting segments follow cinematography rules, the concatenation of segments forms a shorter summary of the long video. The method provides knobs to increase and/or decrease both the video and textual summary length to suit the application. While we evaluate our methods on egocentric videos and TV episodes, similar techniques can also be used in commercial and government applications such as sports event summarization or surveillance, security, and reconnaissance.

Our proposed approach consists of four main components:

1. Identification of interesting segments from the full video;
2. Key frame extraction from these interesting segments;
3. Annotations for these key frames are generated using a deep video-captioning network; and
4. The annotations are summarized to generate a paragraph description of the sequence of events in the video.

3.2.1 Superframe Segmentation Framework

Most work on extracting key segments from video has been done on extracting aesthetically pleasing, informative, or interesting regions. Realizing these key segments will ultimately be stitched, we additionally observe cinematographic rules which prefer segment boundaries with minimum motion. Following Gygli *et al.* [76], each of these key segments are termed superframe cuts.

As videos used in this research are several hours long, every ten frames are first

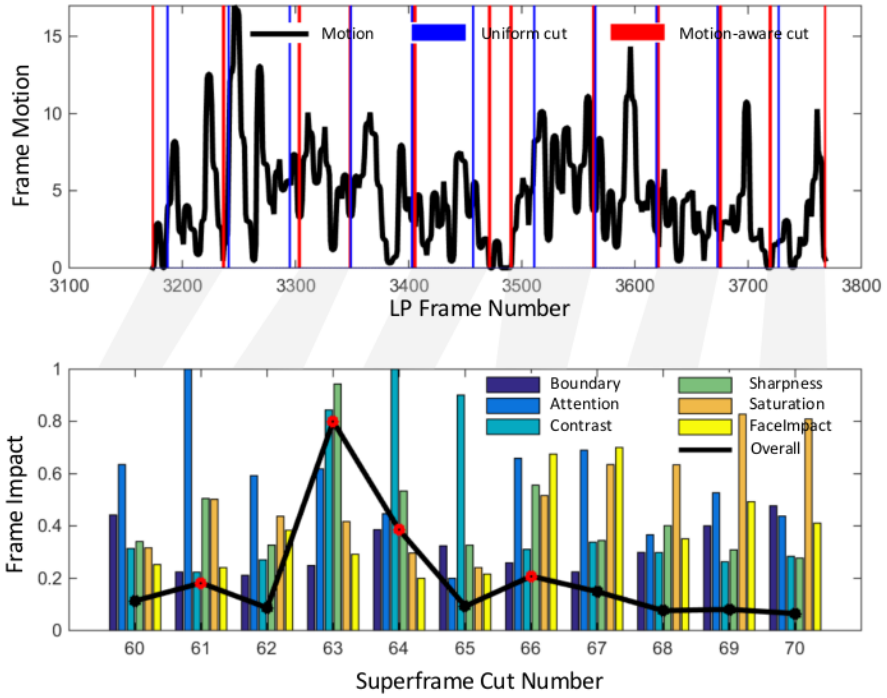


Figure 3.2: (top) The black trace shows frame-to-frame motion, the blue bars show evenly spaced boundaries, and red bars show the final selected superframe boundary cuts. (bottom) The corresponding superframes impact scores as bar graphs, overall interestingness score as a black line, and red pentagrams indicate selected superframes.

averaged. The resulting low pass filtered and shortened video is split into s fixed length segments. Optical flow motion estimates are generated, then using cinematographic rules from Gygli *et al.* [76], the segment boundaries gravitate towards areas of local minimum motion. Figure 3.2 (top) shows eleven superframe cuts from a typical video. The black trace shows the frame to frame motion, the blue bars show the initial evenly spaced segmentation boundaries, and the red bars show the final selected boundary cuts.

Generating Superframe Cut Fitness Scores

Given s superframe cuts, we need to decide which are worthy of inclusion in the final summary, and which will be edited out. Worthiness will be determined by a non-linear combination of scores measuring a superframe cut's fitness regarding Boundary,

Attention, Contrast, Sharpness, Saturation, and Facial impact. Each of these will be described next.

Boundary Score

A Boundary score, B is computed for each superframe region, where the score is inversely proportional to the motion at each boundary neighborhood. Similar to [98], we stack the optical flow between consecutive frames in the x- and y- directions. Motion is computed as $M(t)$ (see key frame selection section below), then given $M(t)$, $B = 1/M(t)$.

Attention Score

Each of these superframe regions are evaluated for aesthetic and interesting properties. Similar to [75][74], an Attention score, based on temporal saliency is first used. The Attention score, A is a combination of the superframe motion, m and variance, v, where m and v correspond to the mean and variance of all non-boundary frames motion in a superframe cut. The final Attention score $A = \alpha * m + (1 - \alpha) * v$, with $\alpha = 0.7$.

The measures of Contrast, Sharpness, Colorfulness, and Facial impact are computed for all frames in each superframe cut and then averaged to report four values for each superframe cut.

Contrast Score

Similar to [85], a Contrast score is computed. To calculate the Contrast score, C, each frame in a superframe cut is converted to luminance, low pass filtered, and resampled to $64 \times width$, where 64 is the new height and $width$ is selected to preserve the aspect ratio of the frame. The Contrast score, C, is the standard deviation of luminance pixels.

Sharpness Score

Similar to [85], a Sharpness score is computed. To calculate a Sharpness score, E, the frames are converted to luminance, then divided up into 10×10 equally spaced regions. Using the center 7×7 regions, the standard deviation of luminance pixels is calculated three times centered on each region, where each of the three times a random shift is added, and the median of the three standard deviation values is reported for each

of the 49 regions. The Sharpness score, E is the maximum of the 49 standard deviation values.

Colorfulness Score

Similar to [86], a Colorfulness score, S is computed. The frames are converted to HSV space, low pass filtered, resampled to $64 \times width$, where 64 is the new height and $width$ is selected to preserve the aspect ratio of the frame, then the mean saturation value from the frame is reported.

Facial Impact Score

Ptucha *et al.* [88] reported on the importance of facial content in imagery, and described a method for generating aesthetically pleasing crops of images containing facial information. Similar to Gygli *et al.* [76], but following the rules from [88], we compute a Face impact score, F which favors larger and more centrally located faces. Each face is assigned an impact score and the sum of all face scores is reported as a Face impact score, F.

To convert from pixels to a universal unit of measure, the size of a face, FS is normalized to the size of the image using:

$$FS = \frac{faceWidth^2}{(imageWidth \times imageHeight)} \tag{3.1}$$

where $faceWidth$ is the width of the face bounding box in pixels, or 2 $\times$ intraocular distance if bounding boxes are not square. Finally, following [88], the face size attribute, FSA is normalized to 0:1, centered on 0.5 for a typical face:

$$FSA = -72.4 * FS^3 + 27.2 * FS^2 - 0.26 * FS + 0.5. \tag{3.2}$$

For the face location, faces centered left-right and just above top-bottom center line are favored. The face centrality attribute, FCA is measured with respect to the 2D

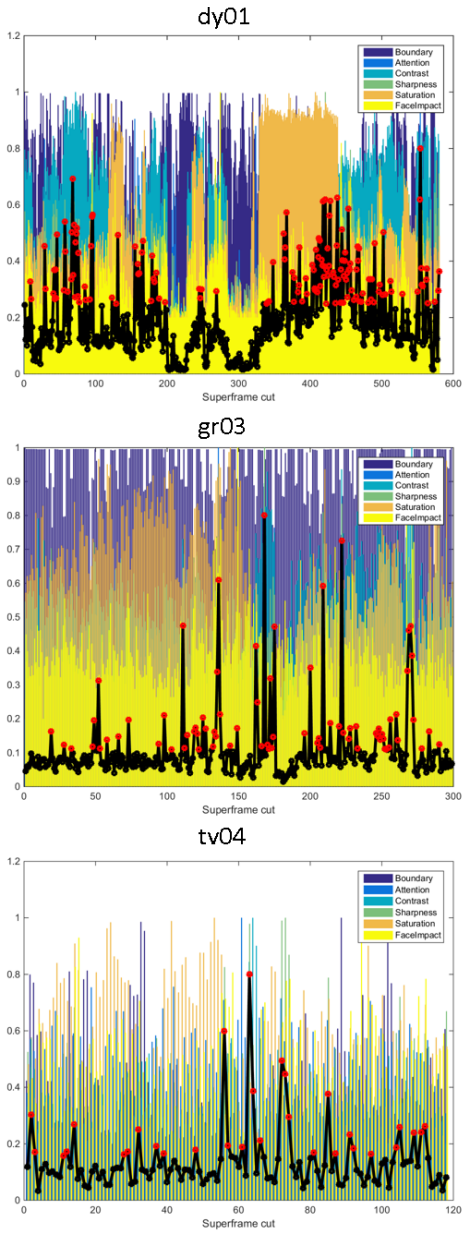


Figure 3.3: Impact scores for superframe cuts in the three test videos. Different colors represent contribution of features- Boundary, Attention, Contrast, Sharpness, Saturation and Face impact. X-axis is the superframe cut number and Y-axis is the normalized impact score. Solid black is I_{score}, red pentagrams show selected superframe cuts with $\omega = 50\%$. (Figure best viewed at 200%).

Gaussian:

$$FCA = \frac{1}{2\pi\sigma_x\sigma_y} e^{\frac{\frac{\delta_x^2}{\sigma_x^2}+\frac{\delta_y^2}{\sigma_y^2}}{x}} \tag{3.3}$$

where:

$\sigma_x = 2 \times imageWidth/3$;

$\sigma_y = imageHeight/2$;

$\delta_x = abs(faceCentroidX - imageWidth/2)$;

$\delta_y = abs(faceCentroidY - 3 \times imageHeight/5)$;

$faceCentroidX$ is the centroid column of the face region; and $faceCentroidY$ is the centroid row of the face region.

For high impact, faces need to have both high FSA and FCA. The face impact score for the entire image, F is $\sum FSA \times FCA$ for all detected faces in the image.

Fusing Scores Empirical testing has shown that Attention (A), Contrast (C), and Sharpness (E) are essential elements to the usefulness and fidelity of a superframe region. After normalization, the product of these three scores are used to form a baseline score for each superframe region. Boundary motion (B), Saturation (S), and Face impact (F) increase this baseline score by $\eta(B + F) + \gamma(S)$, where $\eta = 0.35$ and $\gamma = 0.2$. The final measure of superframe cut interestingness score is computed as:

$$I_{score} = A \cdot C \cdot E + \eta(B + F) + \gamma(S) \tag{3.4}$$

Figure 3.2 (bottom) shows the corresponding superframe segments from Figure 3.2 (top), but with the individualized fitness scores and the overall I_{score} in solid black. After I_{score} is calculated for an entire video, the top superframe cuts (red pentagrams in Figure 3.2 (bottom)) are selected by only using superframe cuts which comprise ω% of the total energy. These selected superframe cuts define the region in the original video which are used for visual and annotation summaries. Video summary duration can be altered by changing ω.

3.2.2 Key Frame Selection

For each selected superframe cut, we use optical flow displacement fields between consecutive frames to identify key frames [99]. A hierarchical time constraint ensures that fast movement activities are not omitted. The first step in identifying key frames is the calculation of optical flow for the entire superframe cut and estimate the magnitude of motion as a function of time. We use an OpenCV implementation [100] of optical flow to estimate motion. The function is calculated by aggregating the optical flow in the horizontal and vertical direction over all the pixels in each frame to calculate a magnitude of motion:

$$M(t) = \sum_i \sum_j \mid OF_x(i,j,t) \mid + \mid OF_y(i,j,t) \mid \tag{3.5}$$

where $OF_x(i,j,t)$ is the x component of optical flow at pixel i,j between frames t and $t-1$, and similarly for y component. As optical flow tracks all points over time, the sum is an estimation of the amount of motion between frames. The gradient of this function is the change of motion between consecutive frames and hence the local minimas and maximas represent important activities between sequences of actions. For capturing fast moving activities, a temporal constraint between two selected frames is applied during selection [101]. Frames are dynamically selected depending on the content of the video. Hence, complex activities or events would have more key frames, whereas simpler ones may have less.

3.2.3 Video Clip Captioning

Video clip captioning is achieved by modifying S2VT [78] with new frame features and introduction of key frame selection. Each key frame is passed through the 152-layer ResNet CNN model [3] pre-trained on ImageNet data, where the $[1 \times 2048]$ vector from the last pooling layer is used as a frame feature. These key frame feature vectors are passed sequentially into a Long Short Term Memory (LSTM) network [102], a recurrent

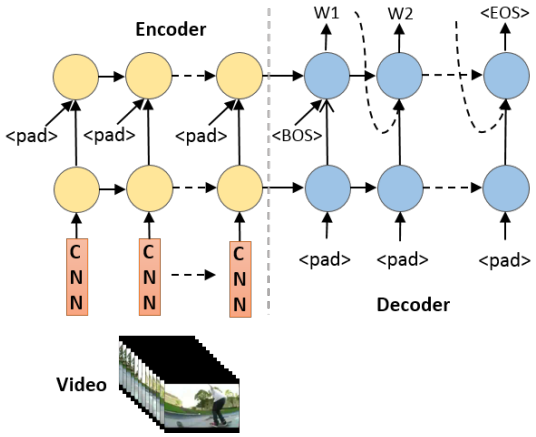


Figure 3.4: S2VT: A two layer LSTM model to learn video representation in the encoder and word representation in the decoder.

neural network approach used in the speech recognition, language translation, as well as visual annotation. The S2VT framework first encodes f frames, one frame at a time to the first layer of a two layer LSTM, where f is of variable length. This latent representation is then decoded into a natural language sentence one word at a time, feeding the output of one time step into the second layer of the LSTM in the subsequent time step.

During training, a video sequence and corresponding text annotation pairs are input to the network. During testing, f key frames around a superframe video segment are encoded into the trained neural network. Once all frames are processed, a begin of sentence keyword is fed into the network, triggering word generation until and end of sentence keyword is produced. The two layer LSTM is fixed to 80 time steps by zero padding shorter sequences and clipping longer sequences. This includes both the input frames for each clip as well as its associated caption.

3.2.4 Text Summarization

The *sumy 0.4.1* python framework along with NLTK libraries were used to evaluate Luhn's algorithm, Edmundson's heuristic method, Latent Semantic Analysis (LSA),

LexRank, TextRank, SumBasic and KL-Sum text summarization techniques. Before passing video clip captions into the text summarizers, duplicate captions were filtered out. The temporal order of each caption was preserved, and the summary length was fixed to 24 sentences for this study, but can be changed to any length greater than the number of input captions.

In order to evaluate the summaries produced in this way, we turned to ROUGE [71], a set of objective metrics of summarization quality that can be calculated automatically, making them ideal for development and comparison of summaries generated by multiple summarization models. These metrics rely on methods of measuring word overlap between the output of a summarization system and one or more human generated reference summaries. Although simple, the ROUGE metrics correlate very highly with human evaluations. Here we use ROUGE-2, which measures the number of bigrams (i.e., two-word sequences) appearing in the summarization output that also appear in the reference summaries. ROUGE-2 is one of the more commonly used variation of the ROUGE metric in the text summarization research community and is the variant of ROUGE-N with the highest correlation with human evaluation. Using Lin's [71] notation, ROUGE-2 is formulated as follows: where $Refs$ is the set of reference summaries, $Count(bigram)$ is the count of a bigram, and $Count_{match}(bigram)$ is the number of matching bigrams in the summarization output:

$$ROUGE2 = \frac{\sum_{S \in Refs} \sum_{bigram \in S} Count_{match}(bigram)}{\sum_{S \in Refs} \sum_{bigram \in S} Count(bigram)} \tag{3.6}$$

3.3 Results and Discussion

3.3.1 Datasets

We demonstrate summarization on the VideoSet [103] dataset. This dataset is comprised of eleven long (45 minutes to over 5 hours) videos in three categories: Disney, egocentric, and TV episodes. Eight videos are used for training and three (DY01, GR03, TV04)

for testing. The captioning model was pre-trained on the training split of the MSVD dataset [39] as the training data form VideoSet is not deemed sufficient.

3.3.2 Captioning Results

Table 3.1 compares ROUGE-2 scores from the ground truth captions and summaries provided with the VideoSet dataset using several text summarization methods. The ground truth annotations for each five/five/ten second segments for the egocentric/Disney/TV videos, respectively, were compared to a single ground truth summary for each video. These results can be considered as the *upper bound* of the summarization methods, which suggest that the LexRank, LSA, and SumBasic methods are generally performing best.

Table 3.1: ROUGE-2 scores (higher is better) for VideoSet dataset. (lu= Luhn, ed=Edmundson, lsa=LSA, tr = text-rank, lr = LexRank, sb = SumBasic)

Video	lu	ed	lsa	tr	lr	sb	kl
DY01	0.32	0.26	**0.42**	0.20	0.29	0.36	0.18
GR03	0.21	0.20	0.22	0.15	0.16	**0.23**	0.16
TV04	0.35	0.14	**0.38**	0.22	0.18	0.16	0.11

After training, text summarization was applied to the three VideoSet test videos: DY01 a 5.5 hour video recorded by a Walt Disney World tourist; GR03 a 3 hour video depicting everyday activities; and TV04 a 45 minute episode of the TV show *Numb3rs*. Table 3.2 indicates strong benefits to using our key superframe segments. The TV04 was the shortest video and the summary contained numerous unique reference to names which cannot be learned from the training set. The summary of this video had numerous character and character usage errors, most likely due to the lack of training data to learn faces and appearances.

3.3.3 Human Evaluations

We created a task in which ten human judges rated our machine generated text summaries for overall summary semantics, sentence syntax, and sentence semantics on a 1

Table 3.2: ROUGE-2 scores for machine generated vs. ground truth on VideoSet test videos. (LSA/LexRank/SumBasic methods)

Test Video	All Clips	Key Clips
DY01	0.25 / 0.17 / 0.21	0.31 / 0.30 / 0.31
GR03	0.15 / 0.07 / 0.14	0.15 / 0.11 / 0.15
TV04	0.02 / 0.02 / 0.02	0.01 / 0.01 / 0.01

Table 3.3: Example of a machine generated summary for DY01 video using LSA. (<en_unk> indicates that the model generated a word representation not found in the trained dictionary.)

I used my phone while waiting for the tram to depart. I looked through the attendant and i rode the tram. My friends and i waited for the tram to depart. My friends and i stood around the tour guide. My friends and i posed for a group picture. My friends and i talked about our day while walking around the park. My friends and i waited in the <en_unk> <en_unk> talking to the theater. My friends and i listened to the tour guide. I talked on my phone while walking around the park. My friends and i talked while moving along the line. I stood with a group of my friends talking. My friends and i walked through a dark room. My friends and i talked about our food while walking around the park. My friend and i talked about the camera while walking around the park. My friends and i talked about our camera while waiting around the park. My friends and i walked with our group leader through the park while talking. I stood in a dark place and talked to my friends. I walked through a dark room talking with my friends. I watched a mascot entertain i waiting. I grabbed some food while moving along the line. My friends and i sat at the table and had dinner. My friends and i waited at the table and had dinner. I watched a mascot entertain another group. My friends and i sat at the table and talked.

(very poor) - 5 (very good) Likert-type scale. The questions asked to the human judges were-

- After reading the summary, would you be able to describe the video to another person.
- Rate the quality of the syntax/grammar of the summary sentences (missing words, word order, incorrect words, unknown words, punctuation, upper/lower case, duplicate words/sentences).

- Rate the quality of the semantics/clarity/understanding of the summary sentences.

Table 3.4: Human evaluation scores on machine generated video summaries using LSA.

	DY01	GR03	TV04
Summary Semantics	3.65	2.35	1.40
Sentence Syntax	3.55	2.40	1.65
Sentence Semantics	3.80	2.35	1.45
Average	3.67	2.37	1.50

For overall summary and sentence syntax, the LSA and LexRank methods were preferred. For sentence semantics, all methods performed comparably. Judges rated the TV04 summaries much lower than DY01 and GR03. Since the TV04 video was a TV episode, the summaries lacked context due to absence of character names. Moreover, compared to daily activity videos, TV episodes have very specific domain that may require large amounts of training data.

3.3.4 Evaluating Superframe Cut Selection

We use the SumMe Dataset [76] to evaluate the effectiveness of our features in superframe cut selection. The SumMe Dataset consists of 25 videos, ranging from one to seven minutes (950 to 9721 frames). An ablation analysis across the six features of Boundary, Attention, Contrast, Sharpness, Saturation, and Face impact was performed across all 25 videos. A five frame averaging filter was used, and then every 10th frame was extracted and resampled so frame width=480 pixels. The mean value for each feature in each superframe cut along with the mean ground truth relevance score was passed into the ablation analysis. A mean squared error from a linear regression model was used as a fitness criterion.

Both the mean rank and top-k ranked columns of Table 3.5 show all features have significant usefulness in superframe cut selection. Although the Contrast and Saturation features have the lowest rank, the top-3 column shows the balanced nature of all the features. While the Boundary feature was an average performer, the human annotators

Table 3.5: Feature evaluation on SumMe dataset. Mean rank position (lower is better); number of times feature was selected 1st; 1st or 2nd; and 1st, 2nd, or 3rd.

Feature	Mean rank	top-1	top-2	top-3
Contrast	2.72 +/- 2.19	7	8	12
Saturation	2.80 +/- 2.16	6	8	10
Boundary	2.92 +/- 1.75	1	6	12
Face impact	2.92 +/- 1.89	1	9	11
Sharpness	3.12 +/- 2.01	3	6	11
Attention	3.24 +/- 2.01	3	7	9

rated each frame independently, not taking into account cinematographic rules. While the Face impact was found to be one of the most important factors in [88], only 12 out of 25 videos contained faces in this dataset. The low performance of Attention is surprising, and follow-on research finds the frame averaging is critical towards achieving high importance of the Attention score. For the SumMe dataset, the six features had an overall RMSE of 0.0271 as compared to the ground truth, showing this suite of features are excellent indicators of frame relevance.

3.3.5 Evaluating Key Frame Selection

We use the Keyframe-Sydney (KFSYD) Dataset [104] to evaluate the motion magnitude based key frame election. This dataset consists of ten videos, each with three independent sets of ground truth frame summaries. Table 3.6 reports the ratio of selected key frames that match with ground truth. A frame is considered a match if it is within n-neighborhood of a ground truth frame. top-k refers to matching k-highest probability frames with ground truth. Results reported in the table are averaged over all videos and all ground truth summaries.

Table 3.6: Evaluation scores for key frame selection. High ratio is better.

top-k	15-neighbor	25-neighbor
top-8	0.50	0.66
top-16	0.54	0.69
top-24	0.60	0.72
top-32	0.60	0.72

Chapter 4

Generative Models

An ambitious goal for machine learning and signal processing research is to be able to represent different modalities of data that have the same meaning with a common latent representation. A sufficiently powerful model should be able to store similar concepts in a similar vector representation or produce any of these realizations from the same latent vector. Successfully mapping visual and textual modality in and out of this latent space would significantly impact the broad task of information retrieval.

Recent success in image captioning [13, 26, 14, 27] has shown that deep networks are capable of providing apt textual descriptions of visual data. In parallel, advances in conditioned image generation [28, 29, 30, 31] provide photo-realistic and diverse images from a text based prior. A common occurrence in the aforementioned domains is the presence of a latent vector representation that facilitates modality transition. In this study, we combine the networks used in these domains by merging the latent representations obtained during transition. The proposed model is called Multi-Modal Vector Representation (MMVR). We demonstrate the efficacy of our model in within-domain and cross-domain transformations.

The rest of this chapter is organized as follows: Section 4.1 reviews related work associated with models using latent representation and introduces the relevant pre-requisites for the entire framework. Section 4.2 describes the architecture and methodology in de-

tail. Section 4.3 discussed the experiments along with results.

4.1 Related Work

The notion of a latent space where similar points are close to each other is a key principle of metric learning. The representations obtained from this formulation generalize well when the test data has unseen labels. Models based on metric learning have been used extensively in the domain of face verification [105], image retrieval [23], person-re-identification [106] and zero-shot learning [107].

4.1.1 Multi-Modal Learning using Vector Representation

Ngiam *et al.* [108] used an autoencoder model to learn cross-modal representations and showed results with audio and video datasets. Srivastava *et al.* [109] used deep Boltzmann machines for multi-modal learning on images and tags. Their formulation could generate tags from images or images from tags. Sohn *et al.* [110] introduced a novel informative theoretical objective that was shown to improve deep multi-modal learning for language and vision. Joint language and image learning based on image category was shown in [111]. They used joint training for zero-shot image recognition and image retrieval. Sohn *et al.* [112] introduced multi-class N-tuple loss and showed superior results on image clustering, image retrieval and face re-identification. Eisenschtat *et al.* [113] introduced a 2-layer bidirectional network with batch-normalization and dropout techniques to map vectors coming from two data sources by optimizing correlation loss. Wang *et al.* [114] learned joint embeddings of images and text using a two branch neural network by enforcing margin constraints on training objectives. Recently, Wu *et al.* [115] leveraged this concept to associate data from different modalities. Our work shares similarities with [115]. However, we focus on generating visual/textual data.

4.1.2 Conditional Image Generation

Generative Adversarial Networks (GANs) [32] are a sub-class of generative models based on an adversarial game. Training a GAN involves two models: a generator that maps a random distribution to the data distribution; and a discriminator that estimates the probability of a sample being fake or real. A GAN produces sharp images but the generated images are not always photo-realistic. To improve upon photo-realistic quality, class category [31, 116, 117], caption [28, 29] or a paragraph [118] has been used to condition image generation. Reed *et al.* [29] encoded text into a vector to condition images, however direct encoding reduces the diversity of generated images. Introducing an additional prior on the latent code, Plug and Play Generative Networks (PPGN) [28] drew a wide range of image types and introduced a conditioning framework that tells the generator what to draw. Our work is complementary to such captioning and generative models as we define a common latent space that allows transitioning within and from modalities.

4.1.3 Sequence-to-Sequence Models

Sequence-to-sequence [18] models encode the inputs one at a time, then decodes one word at a time, using a recurrent neural network architecture. These models have been used in applications such as sentence vector representations [119, 120], visual question answering [121, 122] as well as video captioning [19, 123] that encodes the entire video, then decodes one word at a time. Paraphrasing sentences [124, 125] is another application of sequence-to-sequence models. Our work leverages the paraphrasing application to generate synthetic captions from a single caption to improve the quality of the generated images.

4.1.4 Image Captioning

Recent advances in recurrent neural networks have enabled generation of a natural language description of still images [13, 14, 26, 89]. The extension of this to video can

be done by pooling over frames [77] or utilizing a fixed number of frames [35]. Our model uses an image captioner to add a caption based prior on image generation.

4.2 Proposed Framework

We introduce Multi-Modal Vector Representation (MMVR) to create a unified representation for visual and text modality in latent space. The architecture is inspired by the PPGN [28] model that consists of an image generator and a conditioning model to guide the generator. Given an image or sentence, MMVR performs iterative sampling to generate data in either modality while conditioning on an input. Figure 4.1 provides an overview of the MMVR architecture. The model can be divided into two interdependent modules: an image generator and an image captioner.

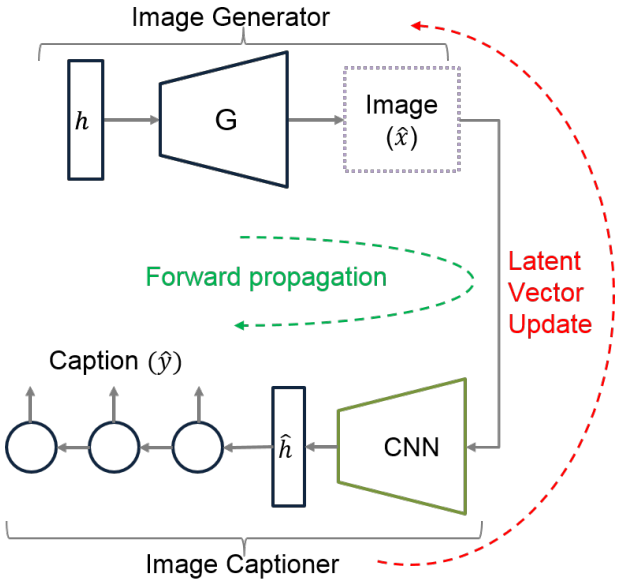


Figure 4.1: Overview of the Multi-Modal Vector Representation model. It consists of two pre-trained modules – an image generator (G) that inputs a latent representation h and generates an image $\hat{x}$; and an image captioner that inputs an image $\hat{x}$ and generates a caption $\hat{y}$. To update the latent vector h, cross-entropy between the generated caption $\hat{y}$ and a ground truth caption y is used while the weights for the generator and CNN are fixed.

The forward pass is initiated by passing a random latent vector h into the image generator which generates an image $\hat{x}$. The image captioner uses the generated image to create a caption. Word-level cross entropy is used to determine the error between the generated caption and a ground truth caption. This error is used to iteratively update h, while keeping all other components fixed. With each iteration, the generated caption approaches the target caption, and the generated image $\hat{x}$ serves as a proxy for the target caption. The gradient associated with the cross-entropy error is specified in (4.1).

$$grad(C) = \frac{\partial \mathcal{L}(C_{pred}, C_{gt})}{\partial h_t} \tag{4.1}$$

Where, $grad(C)$ is the gradient of cross-entropy with respect to latent vector h_t, C_{pred} is the predicted caption and C_{gt} is a ground truth caption. $\mathcal{L}$ is the word level cross-entropy between the two captions.

The $grad(C)$ component of the update rule ensures that the generated images have relevant context. However, to improve the realistic nature of the images, a reconstruction error is included in the update rule. This is computed as the difference between h and $\hat{h}$, where $\hat{h}$ is the fully-connected layer representation of the generated image. This component is referred to as a denoising autoencoder in [28]. h is a 4096-dimensional vector in our experiments to match the output dimension of the fully-connected layer of the CNN. Finally, to add diversity in generated images, a noise term $\mathcal{N}$ is also included. The resulting update rule is a weighted sum of four terms and is described in (4.2).

$$h_{t+1} = h_t + \gamma_1 grad(C) + \gamma_2 R(h_t, \hat{h_t}) + \mathcal{N}(0, \gamma_3) \tag{4.2}$$

Where, $R(h_t, \hat{h_t})$ is the reconstruction error which is computed as difference between h_t and $\hat{h_t}$, $\mathcal{N}$ is Gaussian noise with standard deviation γ_3 and h_{t+1} is the latent vector after the update. γ_1 and γ_2 are weights associated with the gradient of cross entropy and the DAE, respectively. We set γ_1, γ_2 and γ_3 hyper-parameters as 1.0, 10^{-3} and 10^{-17}, respectively. We need noise to make it a proper sampling procedure, but found that infinitesimally small noise produces better and more diverse images, which is to

be expected given that the DAE in this variant was trained without noise. We also observed that if faster mixing or more stable samples are desired, then the γ_1 and γ_2 terms can be scaled up or down together.

The update rule is based upon previous works on latent space interpolation [116, 28, 31]. Our model updates the latent vector h iteratively, which is the input to the image generator, based on (4.2). It also encourages h and $\hat{h}$ to be similar, thereby creating a common latent representation capable of generating both images and sentences. We now discuss the main limitations with using a simple cross-entropy term $grad(C)$ as described in (4.1), and propose two approaches to address these.

4.2.1 n-gram Metric Conditioning

An intrinsic limitation with the model described in Section 4.2 is that the generated caption is compared with a single caption. This causes limitations in cases when the order of words in the generated caption is different or when the captions are different only due to an inserted or a deleted word. The cross-entropy establishes word level correspondences between generated and ground truth captions. For example, consider a case when the generator is conditioned on "a red car", whereas the captioner generates "the car is red". Both the captions are semantically very similar but lack one-to-one correspondence between the words. This may result in unwanted updates of the latent vector h due to high word level cross-entropy. We address this by introducing a n-gram metric in the latent vector update. The metric is responsive to cases when generated and reference captions are semantically similar.

Equation (4.3) describes the update rule when the n-gram metric is used in conjunction with cross-entropy. We compute word level differences and scale it with the n-gram metric between the generated and reference captions.

$$h_{t+1} = h_t + \gamma_1 \frac{\mathcal{F}(C_{pred}, C_{gt})}{n} grad(C) + \gamma_2 R(h_t, \hat{h_t}) + \mathcal{N}(0, \gamma_3) \tag{4.3}$$

Where $\mathcal{F}$ is the n-gram metric. In our experiments, we use the BLEU [68] scores as

n-gram metric. As before, our latent vector h is obtained through an iterative process. The resulting representation is capable of synthesizing either of the two modalities.

4.2.2 Conditioning on Multiple Captions

Another way to overcome one-to-one word correspondences between a predicted and reference sentence is to use semantically similar sentences. Moreover, conditioning image generation on a single caption may lead to generation of images that lack details. We condition the generator on multiple captions to synthesize an image. Multiple captions would increase syntactic variability for the generator to condition on, hence improving the overall image quality.

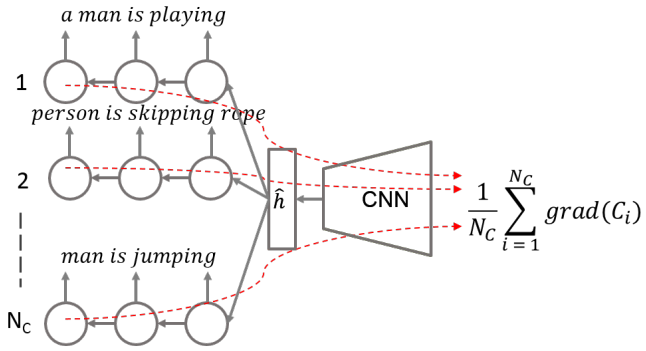


Figure 4.2: Conditioning the image generation through multiple captions by aggregating the gradients from individual caption cross-entropy. Solid black lines show the direction of forward pass during sentence generation and dashed red lines show direction of error back-propagation during latent vector update.

The forward pass is performed in a same way as Section 4.2. The predicted caption is compared against multiple ground truth captions to obtain the individual gradients. The aggregated gradients are used to update the latent vector h. The caption gradient component of the h update rule is replaced by the summation of gradients from multiple captions as shown in (4.4).

$$grad_{avg} = \frac{1}{N_C} \sum_{i=1}^{N_C} grad(C_i) \tag{4.4}$$

Where, N_C is the total number of reference captions and $grad_{avg}$ is the aggregated gradient for all captions. For multiple captions, MMVR uses an image captioning block as shown in Figure 4.2.

4.2.3 MMVR Architecture

Image Generator

The image generator we use is based upon DeePSiM [126] which comprises of three networks:

- an AlexNet [2] CNN encoder. It yields a 4096-dimensional vector.
- an inverted-AlexNet [127] based generator that up-samples the 4096-dimensional vector to an image of size 256×256.
- a discriminator that takes a 256×256 dimensional image and classifies it as real or fake.

Given an input image, the generator is trained to invert the features extracted from a pre-trained AlexNet and reconstruct the input image. The training routine associated with DeepSiM occurs in two steps. First, the CNN based encoder is trained on the ImageNet dataset. The pre-trained CNN is used as a feature extractor to compute the prior h for the generator. During the second training phase, the generator and discriminator are trained using the weighted sum of three losses:

1. The adversarial loss computed using discriminator to determine if image $\hat{x}$ is real or fake.
2. The pixel-wise loss between image x and image $\hat{x}$.
3. The reconstruction loss computed using the pre-trained CNN to compare features associated with image x and $\hat{x}$.

Note that the generator and the discriminator are trained simultaneously but the discriminator is discarded after training and is not part of the MMVR. A limitation of this generator is that it can only generate single object categories. A typical caption would be a description involving multiple object categories. In order to address this issue and improve conditioning on captions, we fine-tune the generator on MS-COCO. Thus, the fine-tuned generator is capable of rendering multiple objects in a image, a characteristic missing in the model trained on ImageNet.

Image Captioner

We use a Long-term Recurrent Convolutional Network (LRCN) [13] which was trained on 82,783 images and 414,113 captions from the MS-COCO dataset [128]. The image captioner is used to steer the search for the 4096-dimensional vector required by the generator to render a representative image for the caption.

Sentence Paraphraser

For the paraphrasing model, we represent the paraphrase sentence pairs as (S_m, S_n). Let s_m denote the word embedding for sentence S_m; and s_n denote the word embedding for sentence S_n. $S_m \in \{s_1...s_M\}$, $S_n \in \{s_1...s_N\}$ where M and N are the length of the paraphrase sentences. As shown in Figure 4.3, the input sentence y generates sentence y_1, y_1 generates y_2, and so on. In our model, we use an RNN encoder with LSTM cells since it is easy to be implemented and performs well on this model. Specifically, the words in S_x are converted into token IDs and then embedded using GloVe [58]. To encode a sentence, the embedded words are iteratively processed by the LSTM cell [18]. Figure 4.3 shows an overview of the paraphrasing model.

There are numerous datasets with multiple captions for images or videos. For example, MSR-VTT dataset [44] is comprised of 10,000 videos with 20 sentences each describing the videos. The 20 sentences are paraphrases since all the sentences are describing the same visual input. We form pairs of these sentences to create input-target

samples. Likewise, MSVD [39], MS-COCO [128], and Flickr-30k [129] are used. Table 4.1 lists the statistics of datasets used. In total, we created around 10M training samples.

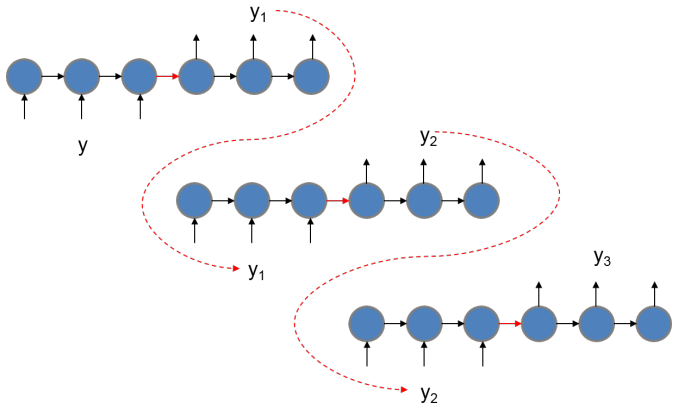


Figure 4.3: The sequence-to-sequence model for generating sentence paraphrases. Both the encoder and decoder process individual elements of their respective sequences in a recurrent manner. The solid black lines show direction of the forward pass and dashed blue lines show the carry-forward of previous element during sequence decoding. <BOS> and <EOS> are special tokens for begin-of-sentence and end-of-sentence, respectively.

Table 4.1: Sentence pairs statistics in captioning datasets.

	MSVD	MSRVTT	MSCOCO	Flickr
#sentences	80K	200K	123K	158K
#sentences/sample	∼42	20	5	5
# sentences pairs	3.2 M	3.8 M	2.4 M	600 K

4.2.4 MMVR Inference

The bi-directional nature of MMVR allows the model to take as input an image or a sentence. The input is then used to condition data generation in either modality. This section describes the transitions between visual and text modalities using the MMVR.

Visual-to-Text – To obtain sentences describing an image, we do a forward pass through the image captioner as described in Figure 4.1. This is the simplest transition

in our model as a pre-trained image-captioner is an independent sub-module of our model.

Text-to-Text – Paraphrasing sentences is achieved through the sequence-to-sequence model present in MMVR. The sequence-to-sequence model as shown in Figure 4.3 was pre-trained on a large corpus of similar sentences for the purpose of paraphrasing.

Text-to-Visual – As described earlier in section 4.2, the image captioner guides the generator during image generation. To this end, we start with a random 4096 dimensional vector h to render an image. The resulting image is captioned using LRCN. The output caption is compared with the ground truth caption and the difference between them is used to modify our h. The process is terminated after 200 iterations and the image rendered by the generator is treated as a representative image for the caption.

Visual-to-Visual – We translate an image into a visually different but semantically similar image. Starting with an image, we generate a caption. Using the sentence paraphraser, we generate a paraphrased caption from the input caption. We then perform the process described in text-to-visual mode to generate an alternate image representation. We employ paraphrased captions to increase diversity in generated images.

4.3 Results and Discussions

We report results for all four modal transformations using MMVR. For clarity, we group the respective transformations into image and text generation sections.

4.3.1 Image Generation

We evaluate image generation task for both the input modalities – visual and text. The qualitative comparisons are aided by quantitative metrics and human evaluations. We show correlations between the inception score [33] that is popular for evaluating generated images, with human evaluations. We also propose a new metric based on object detection that captures the quality of unique objects present in a generated image.

A pre-trained YOLO object detector model [130] is used for this purpose. The model is trained on 80 object categories commonly present in the MS-COCO dataset. We show some examples in Figure 4.4 with synthesized images. Each synthesized image is passed through the object detector model that yields bounding boxes and their corresponding confidences. Formally, *detection score* $= \sum_d \frac{A_d}{A_T} p_d$, which reports the weighted sum of all detections (d) greater than a 0.1 confidence threshold (p_d), where the weight is the ratio of the detected bounding box area (A_d) and the full image area (A_T). Having an area weight is critical since some object detector models may predict a large number of very small bounding boxes. Finally, the reported score is the average over the entire test set comprising of 1000 generated images.

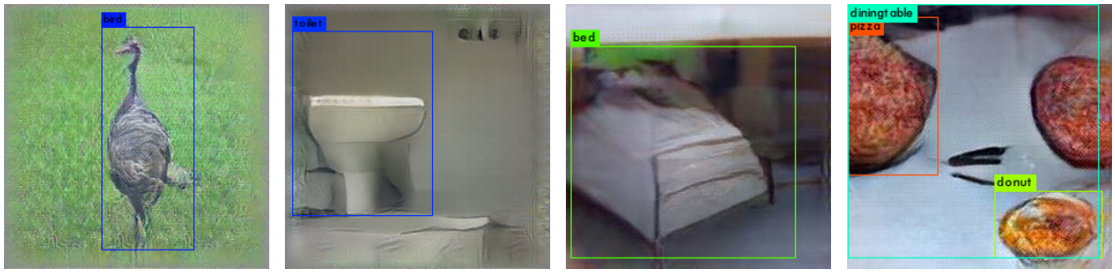


Figure 4.4: Examples of the YOLO object detection on generated images. The bounding boxes and corresponding labels are detections with confidence greater than 0.5 threshold.

Human Evaluations – We conduct human evaluations to validate image generation from PPGN [28] and variants of MMVR. We collected 50 image-caption pairs and asked 80 humans (not including any of the authors) to judge the performance. Each participant was shown eight random images from all methods in random order totaling to 40 samples per person. Each evaluator was asked to rate on a 1 (bad) – 5 (good) Likert-type scale. On average, each method received more than 600 ratings. The questions asked to the human judges were:

- Can you identify any one object in the images?
- How well does the sentence align with the image?

Visual-to-Visual

The visual-to-visual task can be achieved via two different paths using MMVR. Firstly, we encode the image using CNN into FC-6 features and directly input to the image generator. Figure 4.5 (left) show some examples and the results indicate the semantic content in the encoded FC6 features- in terms of spatial location, scale, color, shape, viewing angle, is maintained. For example, while generating the *bird* image, the color of the bird and its beak, viewing angle, and location were all retained even though the representation does not have any spatial context. Other examples also indicate similar trends. Feeding the FC-6 features directly input to the image generator serves as the Baseline method in Table 4.2.

Secondly, MMVR can perform image-to-image transition in a two step process– synthesizing a caption from an input image and using the caption to condition the image generator. We observed that the generated images show more variability through cross modality conditioning. For example, the *bird* image generated through the caption has a independent context from the input image.

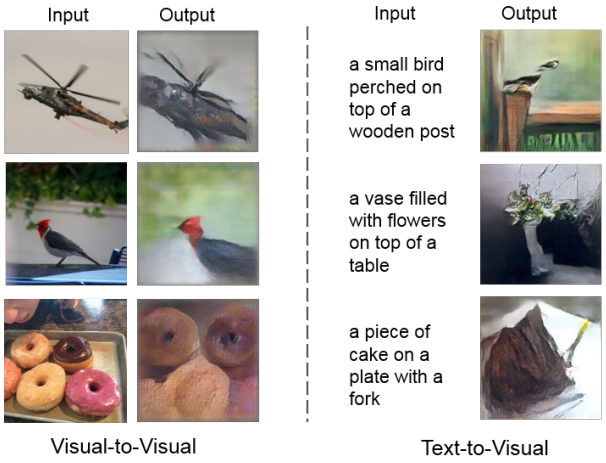


Figure 4.5: Examples of the visual-to-visual (left) and text-to-visual (right) modes of MMVR. The inputs can be the visual or text modalities.

Text-to-Visual

An important property of common latent space is cross-modal transformations. Thus, cross-modal experiments aid in proving that the representations of individual modalities are well aligned in the common space. We show examples of text-to-visual generation in Figure 4.5 (right). It can be observed that MMVR synthesizes reasonable images for captions. As noted in [28], one of the major challenges while conditioning on text include the cross-entropy computation from a sentence with many words. The captions could be 10-15 words long including stop-words which have limited significance on the image content. Moreover, gradients for all words are aggregated and back-propagated, hence significant words may loose importance. This may result in poor image quality. The inclusion of n-gram scaling to the update function and conditioning on multiple ground truth sentences help address such limitations. We observed the captioner generating good captions even for unrealistic images. These could be "fooling" images [30] which are unrecognizable to humans but deep neural networks recognize them with high confidence.

Table 4.2: Evaluation of the generated image quality using the inception, detection and human scores on the test set.

Method	Inception	Detection	Human
Baseline	5.77 ± 0.96	0.762	**2.95**
PPGN [28]	6.71 ± 0.45	0.717	2.34
MMVR (B-1)	7.22 ± 0.81	0.713	2.31
MMVR ($N_c = 5$)	$\mathbf{8.30 \pm 0.78}$	**1.004**	2.71

Table 4.2 compares the text-to-visual technique against the baseline (direct FC-6) and some variations. The inception scores indicate the improvement in generated images when BLEU-1 (B-1) and the multiple caption conditioning ($N_c = 5$) are used. The detection scores for multiple captions are significantly better than other variants. However, BLEU-1 is slightly lower than the baseline result. Our baseline methods got higher human evaluation scores. We believe the reason for this trend is lack of detail in objects generated by multiple captions. The baseline model generates images with

single objects, hence the image are visually appealing.

Conditional Image Generation on Multiple Sentences – To understand the effect of conditioning image generation on multiple sentences, we run experiments by varying the number of sentences. Synthetic sentences were generated using our sentence paraphraser. Figure 4.6 shows the input caption and the generated images with 1, 3 and 5 captions. Image quality enhances with increase in number of sentences. The *food* example also show gains in understanding the concept of quantity (*four*) through text. Similar trends on image quality are observed through the inception and detection score metrics as reported in Table 4.3. The detection score helps prove that multiple sentences assist in generating multiple objects in the image that are recognized by the object detector.

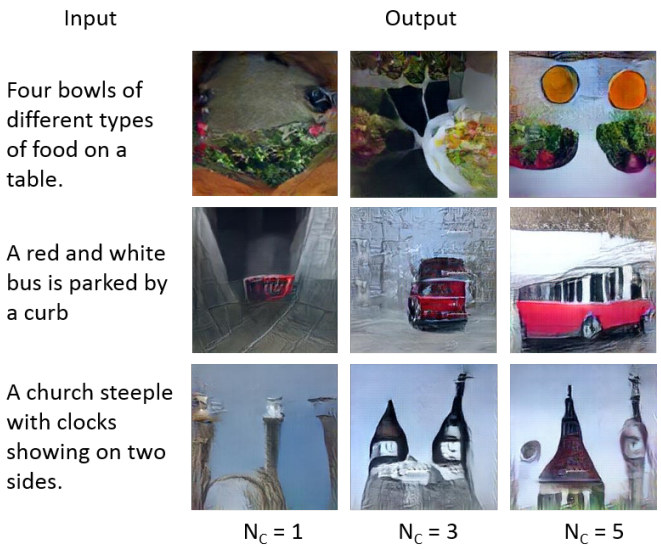


Figure 4.6: Examples of the text-to-image generation as conditioned on varying number of input captions. We observe more detailed images being synthesized with increase in number of captions.

We perform a few ablation experiments to further analyze the text-to-visual mode of the MMVR.

Table 4.3: Evaluation of the generated image quality by conditioning on varying number of paraphrased sentences (N_C).

N_C	Inception	Detection	Human
1	7.22 ± 0.81	0.713	2.31
3	8.04 ± 0.57	0.915	**2.73**
5	$\mathbf{8.30 \pm 0.78}$	**1.004**	2.71

Was the n-gram scaling useful ? – We show examples with and without the n-gram scaling of the gradient term in (3) in Figure 4.7. It is very difficult to judge the two techniques visually. We use only a single caption to condition the image generator to have a fair comparison in this case. The BLEU-1 score was used as the word level error multiplier and it scales the gradients accordingly. The inception scores in Table 4.2 show slight improvement for BLEU-1 against the PPGN.

Which degree n-gram is better for scaling ? – We compare different BLEU scaling in (3) by varying the n-gram metric. Results are reported in Table 4.4. One reason the BLEU-1 performs better than the higher n-gram techniques might be the simple removal of one-to-one word correspondences between the predicted and ground truth captions is sufficient. Higher BLEU metrics require n-gram matching which puts hard constraints on the generated caption. This may cause the significance on important words to be dampened in the overall update.

Table 4.4: Comparison of image quality with different BLEU metrics for scaling the latent vector update function.

Scaling n-gram Metric	Inception Score
BLEU-1	$\mathbf{7.22 \pm 0.81}$
BLEU-2	7.12 ± 0.66
BLEU-3	7.05 ± 0.73
BLEU-4	6.83 ± 0.74

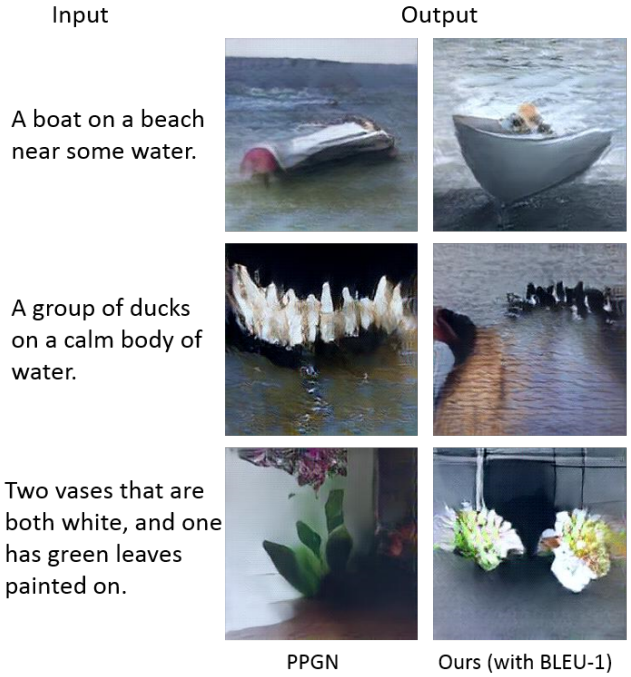


Figure 4.7: Examples comparing the text-to-image for PPGN and the BLEU-1 scaled cross-entropy. Even though slight improvements could be observed with the n-gram scaling, judging the image quality visually is challenging.

Do stop words have significance ? – A caption might have more stop words ("a", "an", "the", "to", etc.) than actual informative words that describe image content. We ran experiments by masking the gradient for the stop words. This did not improve the image quality. We attribute this to the lack of sentence structure after masking stop words. The captioner was trained to generate complete English language sentences. It always generates a complete text description, even though the ground truth caption may be a collection of only relevant words. Hence, for all other experiments we take the running average of the number of words in the caption so all words contribute equally.

Does fine-tuning the image generator help ? – The generator was unable to address common words that occur in a caption (man, woman, person, *numbers*, etc.) since ImageNet does not contain such categories. Moreover, some dominant categories

in MS-COCO dataset like giraffe, stop sign and person are not present in the ImageNet dataset. We visualize the generator results after fine-tuning it on the MS-COCO data. By fine-tuning, the generator is able to semantically capture such categories. Additionally, we observed that multiple objects could also be generated since the original ImageNet model mostly comprised of single object images. An example caption and generated images are shown in Figure 4.8. It could also be interpreted that the generator model correlates better with the captioner since the caption cross-entropy is computed on MS-COCO trained captioner.

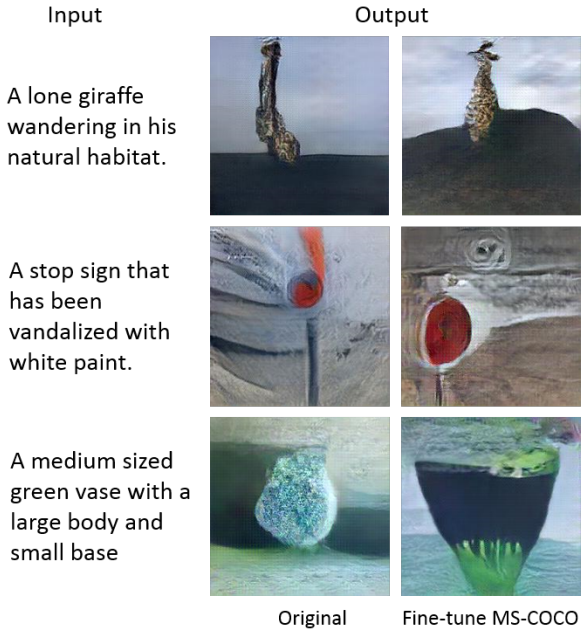


Figure 4.8: Examples that show text-to-image improvements after fine-tuning the generator on MS-COCO dataset. Object categories such as *giraffe* and *stop sign* that are not part of ImageNet dataset show some enhancement in details. We also observed slight improvements in understanding of size, shape and quantity aspects.

4.3.2 Text Generation

Similar to image generation, both input modalities can independently yield text as output. Since we use LRCN [13], the evaluation of the visual-to-text mode is performed

on the test partition of the MS-COCO dataset. We show examples on the left side of Figure 4.9.

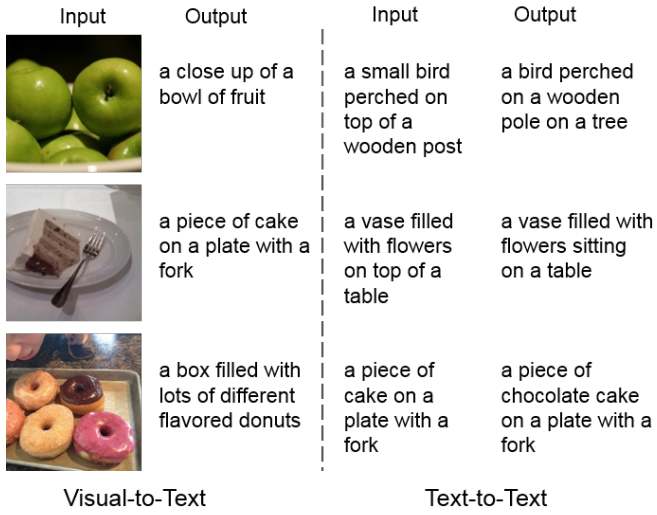


Figure 4.9: Examples of the visual-to-text (left) and text-to-text (right) modes of the MMVR. The inputs can be the visual or text modalities.

We show the usefulness of MMVR in language translation tasks. Given a reference sentence, the objective is to produce a semantically related sentence. We show examples of paraphrasing on the right side of Figure 4.9. Furthermore, to test the robustness of the sentence paraphraser, we run experiments by varying noise levels in the latent space. To evaluate the quality of generated captions, we use BLEU [68], METEOR [69], CIDEr [70] and ROUGE [71] natural language metrics. Since every sample from MS-COCO dataset consists of five captions, we use one of the captions as the input to the paraphraser and the remaining four captions for evaluation. The input caption is fed in the encoder to obtain a vector representation. This representation is corrupted using random uniform noise before being input to the decoder. The results are reported in Table 4.5, where the scale is the noise multiplier. A scale of 0.0 is equivalent to feeding the latent vector without any noise and could be considered as the upper-limit of the paraphraser. We observe that the model is robust to noise up to 1 standard deviation but the performance degrades significantly beyond that. This also indicates that the sentences do not form

very dense clusters in the latent space.

Table 4.5: Evaluation of Text-to-Text paraphrasing model with variation of noise in the latent vector space. The noise scale is the multiplier for the standard deviation of the feature space to generate random uniform noise. Noise with scale 0.0 could be considered as the upper-limit of the paraphraser.

Standard Deviation	0.0	0.1	0.5	1.0	2.0	3.0
BLEU-1	**0.71**	0.71	0.7	0.68	0.56	0.3
BLEU-2	**0.53**	0.53	0.52	0.5	0.38	0.15
BLEU-3	**0.38**	0.38	0.38	0.35	0.24	0.07
BLEU-4	**0.27**	0.27	0.27	0.24	0.16	0.04
METEOR	**0.24**	0.24	0.24	0.23	0.18	0.09
ROUGE	**0.52**	0.52	0.52	0.5	0.42	0.24
CIDEr	**1.03**	1.03	1.02	0.92	0.59	0.15

Vector Arithmetic in Latent Space – Lastly, we evaluate the text-to-text model by performing arithmetic operations in the latent space. Vector arithmetic for language has been shown with words [58] but is still in a nascent stage for complete sentences. The input sentences are fed in the encoder to obtain vector representations. A composite vector is obtained after performing simple mathematical operations on the vector and is fed to the decoder to generate a sentence description. Examples are shown in Figure 4.10. The first three samples demonstrate simple additive properties. Samples 4 and 5 validate more complicated operations and show relationships between objects and actions in the latent space.

1. 'a giraffe is standing' + 'a tall tree in a field' = 'a giraffe is eating leaves off the tree'
2. 'a man is standing' + 'a kite is flying' = 'a man flying a kite'
3. 'two woman are talking' + 'a car is on the road' = 'two ladies discussing car safety'
4. 'a television is on in the living room' + 'a person is sitting' = 'a family living room with couches coffee table and television'
5. 'a piece of cake on a plate with a fork' + 'a man and a woman' = 'a couple of people that are holding a cake'
6. 'a giraffe is standing next to a tree' - 'a tree in a field' + 'a pile of rocks' = 'a giraffe standing next to a pile of rocks'

Figure 4.10: Examples of arithmetic operations in the latent space for the text-to-text model.

Chapter 5

Cross Modal Retrieval

In Chapter 4, we developed methods for bi-directional translation between visual and text modalities. We focused on the generative aspect and discussed the associated challenges. In this chapter, we show the use of common vector representations of different modalities and apply it for cross-modal retrieval. Despite great progress, the generic connection of various written and visual modalities remains challenging. The relationship between multimedia and vectors is further explored in this chapter. The ultimate goal is to discover a common latent representation for different types of sources, as shown in Figure 5.1. In other words, given an input data in any of the following forms: image, audio, video, word, sentence, paragraph, three dimensional model; the framework would encode the input into a semantic vector and decode it to any type of multimedia.

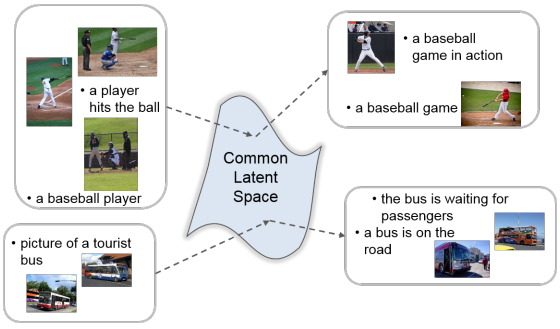


Figure 5.1: Overview of the bidirectional image-text retrieval model.

Figure 5.2 gives an overview of our methodology across several modalities. Latent representations of each modality are mapped to a Common Vector Space (CVS) using individual encoding networks. Similar to the International Color Consortium's device independent profile connection space for color management [131, 132] which maps all inputs to a common reference color specification, a source independent vector connection space requires each new modality to define a single transformation into this reference space. Given M modalities, this architecture only requires M transformations for encoding into CVS. Further, as new modalities are introduced to this common vector space, transformations for existing modalities remain unchanged. Not only is this a significant time savings in the generation of new models, it enables intuitive interaction of data across diverse domains.

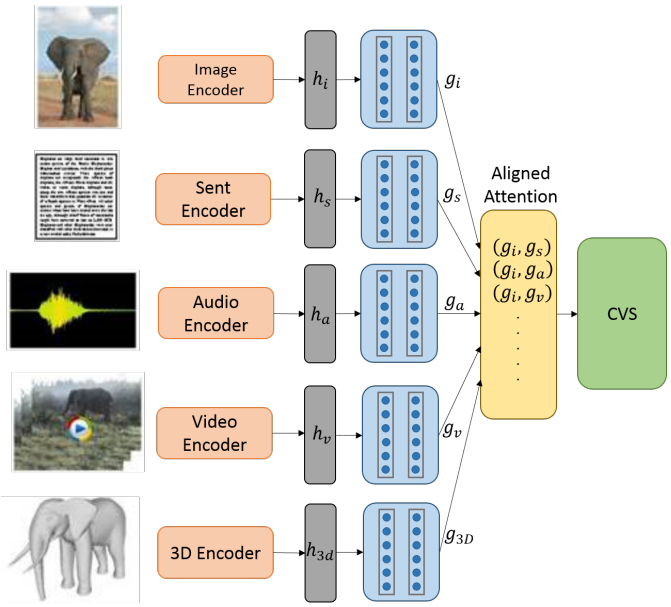


Figure 5.2: The Common Vector Space (CVS) model. Inputs from multiple modalities are mapped to a common latent representation using a series of embedding layers. The red, blue and yellow boxes indicate individual modality encoders, embedding functions and the proposed aligned attention, respectively. The outputs of the attention layer are treated as the common vector representation.

In this chapter, we introduce a novel attention mechanism to align multi-modal

embeddings which are learned through a multi-modal metric loss function. We evaluate the learned common vector space on multiple image-text datasets- Pascal Sentences, NUS-WIDE-10k, XMediaNet, Flowers and CUB. We extend our method to five different modalities and demonstrate cross-modal retrieval on the XMedia dataset. We obtain state-of-the-art cross-modal and zero-shot retrieval across all datasets.

Given adequate sample points, and using appropriate metric learning curriculums, models generalize well to unforeseen data. Methods based on metric learning have been used for face verification [105], image retrieval [23], person-reidentification [106] and zero-shot learning [107]. Inspired by these methods as well as prior approaches extending metric learning to multiple categories [112] and multiple modalities [115], we extend metric learning to both arbitrary number of categories and modalities.

Attention models have been shown to be useful for temporal decoding in language translation [35], image captioning [27], and visual question and answering [133]. This concept allows the decoder to selectively emphasize individual features in the encoder stream. Another form of attention was presented recently for machine translation that does not rely on recurrence or convolutions [134]. We formulate a new concept of attention that aligns latent representations from different modalities. Unlike attention in recurrent models, this concept is applicable to a much broader class of use cases, while boosting model performance significantly.

The main contributions of this chapter include:

- We formulate an efficient vector space model using neural embeddings that act as a bridge between multiple modalities which is easily expandable to new modalities.
- Introduce a novel aligned-attention layer that encourages similar concepts across modalities to have highly correlated latent vectors.
- To the best of our knowledge, we are the first to extend the concept of a common latent representation to several (five) modalities.
- Demonstrate state-of-the-art cross-modal retrieval results on Pascal Sentences,

NUS-WIDE-10k & XMediaNet datasets and state-of-the-art zero-shot retrieval results on CUB & Oxford Birds datasets.

5.1 Related Work

Deep learning has enabled dramatic advancement in image, video and text understanding. For example, image classification [2, 135, 136, 22], object detection [6, 137], semantic segmentation [9, 138], image captioning [13, 27], and localized image description [12] tasks have witnessed tremendous progress in the last few years.

Similarly, our understanding of latent representations of concepts and multi-modal architectures has experienced significant growth. For example, Srivastava *et al.* [109] used deep Boltzmann machines for multi-modal learning on images and tags. Their formulation could generate tags from images or images from tags. Sohn *et al.* [110] introduced an informative theoretical objective that was shown to improve deep multi-modal learning for language and vision. Joint language and image learning based on image category was shown in [111]. They show the use of the joint training for zero-shot image recognition and image retrieval. Ngiam *et al.* [108] used an auto-encoder model to learn cross-modal representations and showed results with audio and video datasets. Sohn *et al.* [112] introduced multi-class N-tuple loss and showed superior results on image clustering, image retrieval and face re-identification. Song *et al.* [139] introduced lifted structured loss, which expanded the N-tuple loss concept such that each positive pair compares distances against all negative pairs.

There have been numerous works on building a joint embedding space between images and captions. Feng *et al.* [140] used correspondence autoencoders to find correlations between images and text. Qi *et al.* [141] used a combination of triplet and contrastive losses to better align objects of the same category from inputs of different modalities. Wang *et al.* [114] learned joint embeddings of images and text by enforcing margin constraints on training objectives. Recent works by [141, 142, 143] leverage

pre-trained architectures for multi-modal retrieval. Qi and Peng [144] performed bidirectional translation process using reinforcement learning to achieve cross-modal retrieval between image and text. The effectiveness of a cross-modal retrieval architecture can be seen by performing zero shot learning as reported by [145, 111].

Often samples have multiple overlapping categories which poses unique challenges for network architecture and loss functions. Vendrov *et al.* [146] used margin based ranking loss with order violation penalty as the distance metric to bring similar image and captions closer in the embedding space. Eisenschtat *et al.* [113] introduced a bidirectional network to map vectors coming from two data sources by optimizing a correlation loss. Lee *et al.* [147] proposed an attention mechanism to compute overall similarity of an image and caption as an aggregate over image regions and individual word outputs of an RNN. Huang *et al.* [148] proposed to use a gated fusion unit to combine the local and global context of an image into a single vector representation. This vector was then matched with the sentence embeddings to bring similar captions closer. In this work, we focus on categorical datasets like [149, 1, 111, 150] where each sample of a modality contains no more than a single category. Additionally, we deploy a novel attention mechanism to align representations from different modalities which learns in a pair-wise fashion.

5.2 Our Model

This section describes the main components of the Common Vector Space (CVS) model.

5.2.1 Embedding Space

Figure 5.2 describes the high-level architecture. Each modality has a unique encoding stage (i.e. image2vec, sent2vec, ...) producing a vector representation, $h_i, h_s, h_a, h_v, h_{3d}$ for image, sentence, audio, video and 3D models, respectively. Each of these vector representations are passed through two embedding functions, the first containing modality-

specific weights, and the second containing shared weights. After passing through the second embedding function, concepts from different modalities are mapped into the CVS. To add a new modality, such as keywords or depth maps, all remaining weights are unchanged, and a set of modality specific encoder and embedding function are introduced.

During training, two input samples are selected at random. The two samples are propagated through the encoder, modality specific embedding layers and the shared embedding layers. While iterating through input pairs, all the layers are learned for their respective inputs. The loss for this pair is used to update the shared and the respective embedding layers. We use a multi-modal metric loss for the input pair. Since every sample is labeled as one of multiple categories, the class labels are used to form positive and negative training pairs. With weights learned, we perform standard cross-modal retrieval and zero-shot retrieval during inference.

Loss Function – One of the important aspects of representing multiple modalities in a shared CVS is to form positive (similar class or concept) and negative (dissimilar class or concept) cross-modal pairs. The inclusion of positive and negative pairings during training ultimately ensures the model can discriminate the data during inference. Positive pairing is done through combinations of samples of the same concept/category. Negative pairs are formed between samples of differing concept/category. Creating relevant positive and negative pairs plays a critical role while learning a multi-modal embedding. For simplicity, we define the loss formulation with just two modalities- image and text, and create positive pairs in three pair formats- between image and image, between text and text and between image and text. Given a set of aligned image-text pairs as training data, the goal is then to learn an image-text compatibility distance $d(f_i, f_s)$ to be used at test time. The distance $d(f_i, f_s)$ is defined between two embedding vectors, f_i and f_s as $||f_i - f_s||_2^2$.

Many recent approaches have explored metric learning functions for mapping the input modalities into a common space. Triplet loss [105] minimizes the distance between

positive samples as compared to negative samples in the CVS. Triplet loss introduces the concept of an anchor point from which positive and negative samples can be compared. We propose to extend the triplet loss formulation to multiple modalities by forming pairs within and across respective modalities. The weighted multi-modal triplet objective is minimized by (5.1).

$$\begin{aligned} L_m = \sum_{i,s} \Big(& \gamma_1 \sum max(0, \alpha_1 + d(f_i^a, f_i^+) - d(f_i^a, f_i^-)) \\ & + \gamma_2 \sum max(0, \alpha_2 + d(f_s^a, f_s^+) - d(f_s^a, f_s^-)) \\ & + \gamma_3 \sum max(0, \alpha_3 + d(f_i^a, f_s^+) - d(f_i^a, f_s^-)) \Big) \end{aligned} \tag{5.1}$$

where, (f_i^a, f_i^+) indicates an embedding pair of the same modality of matching categories with respect to an anchor, whereas (f_i^a, f_s^-) indicates an embedding pair of different modality of mismatching categories with respect to an anchor. γ_1, γ_2 and γ_3 are the weights for within and across modality loss terms. α_1, α_2 and α_3 are the respective distance margins which ensure the distance between positive and anchor points are closer than the distance between negative and anchor points by at least a $margin$.

Architecture

The general architecture is shown in Figure 5.3. It includes five different branches (only image and sentence shown for brevity), each corresponding to a modality. All branches convert the input modality into a vector representation using an encoder function that works as a feature extractor. For example, a pre-trained CNN is used as an encoder for images. The encoder is followed by embedding functions that are unique to each modality. We use a series of three fully connected layers with *tanh* activations for this embedding (1024−512−512). The individual embedding functions are followed by an aligned attention layer that has shared weights for the input modality pair. The output of the attention later is the CVS representation for all input modalities.

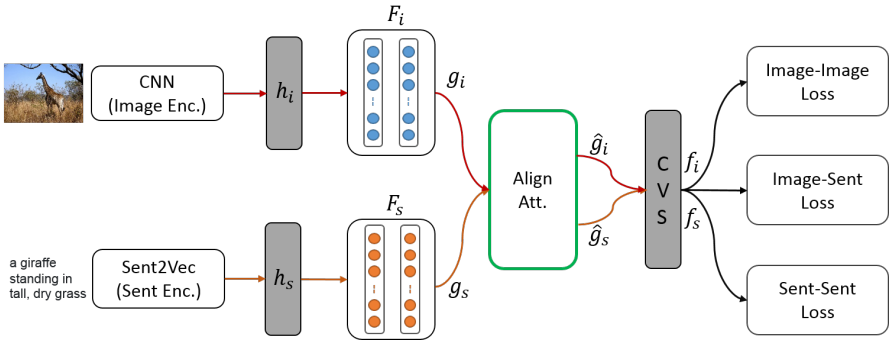


Figure 5.3: Introduced architecture for learning multi-modal embeddings. Only two modalities- image and text are shown for simplicity. Features are extracted from raw inputs using respective encoders (CNN for image and sent2vec for text). Individual embedding functions and shared aligned attention layers are learned during training using positive and negative pairs.

5.2.2 Aligned Attention Layer

A common method for localizing relevant features uses attention models. Soft attention uses a weighted combination of all input features, where the weights are influenced by a recurrent sequence decoder. Soft attention has been used in the context of image and video captioning. Specifically, it computes a feature relevance score $e_i^{(t)}$ for each of the i^{th} features in $v_1, v_2, ..., v_n$ at each decoder time step t.

$$e_i^{(t)} = \mathsf{w}^\top tanh(W_a h_{t-1} + U_a v_i + b_a) \tag{5.2}$$

where, h_{t-1} is the hidden state at the previous time step of the decoder, v_i is the i^{th} feature vector representation, and w, W_a, U_a, b_a are learned parameters. This can be interpreted as an alignment between the encoder and decoder sequences. It allows the encoder to selectively emphasize relevant features based on decoder feedback. The attention vector is normalized using a softmax function as:

$$\alpha_i^t = \frac{exp(e_i^t)}{\sum_{j=1}^{F} exp(e_j^t)} \tag{5.3}$$

The resulting feature is a weighted combination of F input features.

$$\Phi_t(V) = \sum_{i=1}^{F} \alpha_i^t v_i \tag{5.4}$$

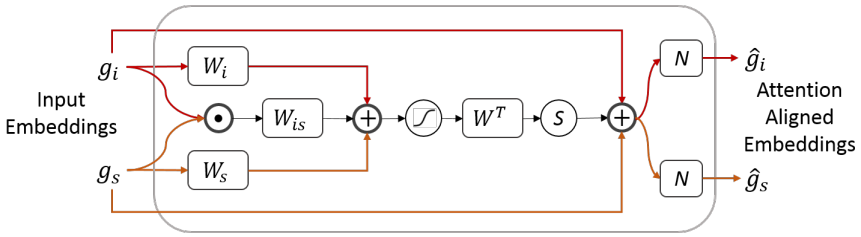


Figure 5.4: Architecture of the aligned attention layer. S is Softmax activation and N is normalization.

Such an additive form of attention is not directly applicable to single embedding vectors. In the case of CVS representations of image and sentence inputs, we have g_i and g_s as the respective embeddings. For any positive input pair, the embeddings should have very high alignment.

$$e^{is} = \mathsf{w}^\top tanh(W_i g_i + W_s g_s + W_{is} g_i \cdot g_s + b) \tag{5.5}$$

where, $g_i \cdot g_s$ indicates element-wise product, which is an indicator of alignment similar to autocorrelation in signal processing. Also, note that $e^{is} = e^{si}$. Similar to (5.3), the resulting vectors are normalized to obtain the attention vectors α^{is}.

$$\alpha^{is} = Softmax(e^{is}) \tag{5.6}$$

We further employ residual connections around each of the embeddings followed by batch-normalization.

$$\hat{g}_i = \alpha^{is} \cdot g_i + g_i \tag{5.7}$$

$$\hat{g}_s = \alpha^{is} \cdot g_s + g_s \tag{5.8}$$

Overall, this can be seen as a multiplicative form of attention since the alignment is obtained using a product of the embedding vectors. The final outputs are l_2 normalized and treated as the final CVS representation. Once in CVS, similar/dissimilar concepts map close/far and it is difficult to tell from which modality a concept was originated.

The proposed attention mechanism is applied to input pairs (two modalities) and is easily extended to additional modalities by including the corresponding weights and the multiplicative terms in (5.5). For example, adding a third audio modality would yield three pair wise attention terms- e^{is}, e^{ia} and e^{sa}, where i, s and a stand for image, sentence and audio. In the resulting three attention equations, the weights corresponding to the first two terms in (5.5), W_i, W_s and W_a are shared. In our experiments, we have extended the attention up to five modalities in a similar fashion.

5.3 Results and Discussion

5.3.1 Datasets

We evaluate our method on multiple datasets which are briefly described in this section. The train/test splits used for all datasets are adopted from previous works.

Pascal Sentence dataset [151] is a collection of image-text pairs in 20 different categories. It contains a total of 1,000 images and each image has five independent sentences. Following [140], we use 800 image/text pairs for training and remaining for testing.

NUS-WIDE-10k is a subset of the NUS-WIDE [149]. Following [149], 10 categories are selected to obtain 1,000 image/text pairs per category. We use 8,000 samples for training (800 per category) and the remaining 2,000 for testing (200 per category).

XMediaNet dataset [152] is a large scale dataset with image-text pairs in 200 different categories. There are 32,000 train and 8,000 test samples for each modality.

Caltech-UCSD Birds (CUB) contains 11,788 bird images from 200 different categories. Each image has ten different sentence descriptions as collected by [111]. Following [111], the data is split into 150 categories into train and validation and the remaining

50 categories for test.

Oxford Flowers-102 (Flowers) dataset contains 8,189 flower images from 102 different categories. Similar to CUB, each image has 10 different sentence descriptions collected by [111]. The images in Flowers are split into 82 training + validation and 20 test classes based on [111].

XMedia dataset [150, 1] is used to demonstrate the applicability of the model for multiple input modalities. The XMedia dataset has five modalities- image, text, audio, video and 3D models. Each sample is labeled in one of twenty categories. XMedia dataset statistics are tabulated in Table 5.1. Readers are referred to [1] for details of individual modality encoders.

Table 5.1: XMedia dataset [1] statistics.

Modality	#Train	#Test	Feature dim. (Method)
Image	4000	1000	4096 (CNN)
Text	4000	1000	3000 (BoW)
Video	969	174	4096 (C3D-CNN)
Audio	800	200	29 (MFCC)
3D Model	400	100	4700 (Light Field)

5.3.2 Implementation Details

We use TensorFlow [153] to train and test the CVS models. All experiments are trained for 50 epochs and use a batch size of 128. The margin hyper-parameters (α_1, α_2, α_3) for the metric loss are 1.0, whereas, the weights (γ_1, γ_2, γ_3) are (0.25, 0.25, 0.5). Adam optimizer is used during training. The learning rate is 1×10^{-3} and we use decay parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$) as reported in [65]. The common vector space is 512-dimension. For the experiments with the category loss, we use a combination of the metric loss and the category loss as the total loss for common representation learning, $L = L_m + L_c$.

Feature extraction − For sentence encoder of Pascal Sentences, CUB, and Flowers datasets, we use the pre-trained skip-thoughts model [119] that yields a 4800-dimension

vector for each sentence. The images are encoded using a ResNet-152 model [136] pre-trained on ImageNet dataset giving a 2048-dimension vector for each image. For the XMedia and XMediaNet datasets, we use the pre-extracted features as provided by [150, 1] and [152], respectively. Similarly, for NUS-WIDE-10k dataset, we use the pre-computed image and sentence features as provided by [149].

Inference – During cross-modal testing of image and sentence retrieval, each image is evaluated against all other sentences. This forms a similarity matrix which is used to retrieve closest sentence samples to an image and vice-versa. While evaluating multi-modal retrieval, the same strategy is utilized for all modality pairs.

5.3.3 Evaluation Metrics

We refer to image-to-text and text-to-image as cross-modal whereas experiments with five different modalities are referred to as multi-modal. We report the mean Average Precision (mAP) scores as described in [140] on all datasets. The Average Precision (AP) is computed for every query on the first R top-ranked retrieved data samples:

$$AP = \frac{1}{M} \sum_{r=1}^{R} p(r) \cdot rel(r) \tag{5.9}$$

where, M is the number of relevant data samples in the retrieved results, $p(r)$ is precision at r, and $rel(r)$ is a binary indicator of relevance of a given rank (one if relevant and zero otherwise). The retrieved data is considered as relevant if it has the same semantic label as the query. mAP is obtained by averaging AP of all queries. As per other works, we report mAP@50 ($R = 50$) for all experiments.

5.3.4 Experiments

Cross-Modal Retrieval

Cross-modal retrieval results are reported in Table 5.2. We show recent reported scores on the Pascal Sentences, NUS-WIDE-10k and XMediaNet datasets. For both text-to-

image and image-to-text retrieval, our method shows clear improvements. Our results are obtained with the CVS model with attention, shared layers and combination of metric and category losses. On the Pascal, NUS-WIDE, and XMediaNet datasets, our method reports a mean average precision on image-to-text and text-to-image retrieval improvement over the next best previous works by $[3\%, 1\%, 7\%]$ and $[7\%, 8\%, 8\%]$ respectively. The presence of jointly learned weights in the attention and shared layers makes the representation symmetric across different modalities for unseen test samples. We observe most improvements in text-to-image, probably due to the maturity of captioning research. A major challenge with the NUS-WIDE-10k dataset is that each image has only one text sample per image. The Pascal Sentences dataset has multiple ground truth captions per image which makes the common vector space more robust for retrieval. Some sample retrieval visualizations are shown in Figure 5.9.

Zero-Shot Retrieval

To test the robustness and generalizing ability of our CVS model, we evaluate performance on data categories that are not part of the training set (zero-shot retrieval). This presents a more challenging retrieval setting for the CVS model. To allow for a direct comparison with previous reported results, we follow the evaluation strategy from [111] to compute the average precision. The average prevision @50 for image-to-text retrieval is the ratio of the top-50 scoring images whose class matches that of the text query, averaged over the test classes.

Zero-shot retrieval results on the CUB and Flowers datasets are reported in Table 5.3. The results with our CVS model show consistent improvement as compared with previously reported state-of-the-art results. On the CUB and Flowers datasets, we report the mAP improvements of 7% and 4% above the next best reported results respectively. The combination of the two modalities through the aligned attention assist in forming a robust CVS that performs well on unseen categories. The attention helps capture the semantic differences among the classes which is very challenging since both of these

Table 5.2: Mean average precision for cross-modal retrieval (image-to-text and text-to-image) on Pascal Sentences, NUS-WIDE-10k and XMediaNet datasets. AA is Aligned Attention.

Dataset	Method	Img2Txt	Txt2Img
Pascal Sent.	UNCSM[141]	0.304	0.282
	Deep-SM[142]	0.446	0.478
	ACMR[143]	0.535	0.543
	DCKT [154]	0.582	0.587
	MCSM [155]	0.598	0.598
	CBT [144]	0.602	0.583
	Baseline CVS	0.589	0.563
	Baseline + AA	**0.639**	**0.650**
NUS-WIDE	UNCSM[141]	0.312	0.354
	Corr Full AE[140]	0.331	0.379
	CSGH [156]	0.542	0.569
	ACMR[143]	0.544	0.538
	DCKT [154]	0.556	0.584
	Baseline CVS	0.439	0.485
	Baseline + AA	**0.566**	**0.669**
XMediaNet	CBT[144]	0.516	0.464
	CM-GAN[157]	0.521	0.466
	Baseline CVS	0.536	0.495
	Baseline + AA	**0.598**	**0.546**

datasets are fine-grained in nature.

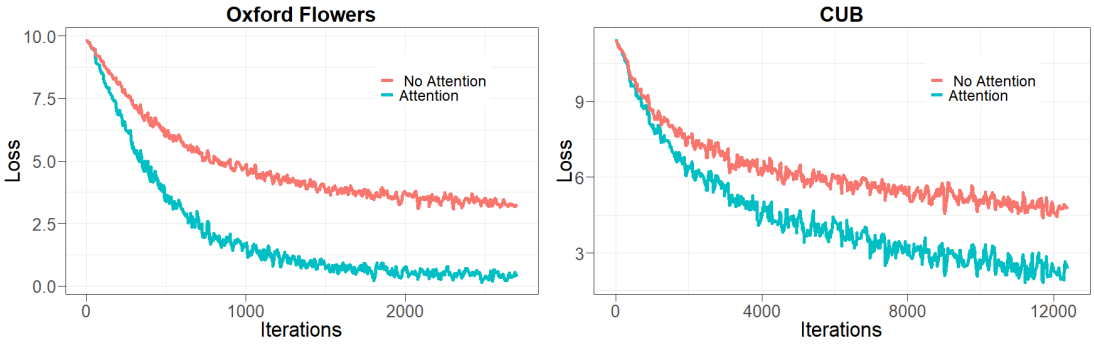
We noted the alignment of the embedding of the two input modalities makes the task of learning a common embedding easier. As an example, Figure 5.5 shows the loss curves with and without our aligned attention layer for the Flowers and CUB datasets. All the hyper-parameters are identical for both experiments within each dataset. The loss curves clearly show faster and improved learning with inclusion of the aligned attention layer.

Multi-Modal Retrieval

To demonstrate the suitability of CVS to multiple modalities, Table 5.4 reports the mean average precision scores on the XMedia dataset which has five modalities. We evaluate

Table 5.3: Mean average precision for zero-shot retrieval on CUB and Oxford Flowers datasets. Baseline CVS model is trained with only multi-modal metric loss, AA is Aligned Attention. The best models reported include classification loss.

Dataset	Method	mAP@50
CUB (Birds)	DA-SJE [111]	0.368
	DS-SJE [111]	0.500
	Hubert [158]	0.476
	Cosine [159]	0.500
	Dorfer [145]	0.522
	Baseline CVS	0.538
	Baseline + AA	**0.589**
Flowers	DA-SJE [111]	0.459
	DS-SJE [111]	0.596
	Cosine [159]	0.602
	Dorfer [145]	0.640
	Baseline CVS	0.604
	Baseline + AA	**0.679**



(a) Oxford Flowers dataset.

(b) CUB (Birds) dataset.

Figure 5.5: Loss curves for models with and without the aligned attention layer.

the retrieval of every modality against all the other modalities. For the XMedia dataset, the number of samples across modalities is highly imbalanced (Table 5.1) and the encoder features dimension of the audio modality is very low which makes it hard to converge with other inputs. We believe that having a balanced dataset, improved features and modality specific curriculum learning would be helpful for such problems.

Table 5.4: Mean average precision for multi-modal retrieval on XMedia dataset (I - image, T- text, A - audio, V - video, 3D - three dimension). Q is query and R is retrieval modality.

Q \ R	I	T	A	V	3D
I	–	0.908	0.708	0.801	0.731
T	0.950	–	0.743	0.828	0.769
A	0.416	0.477	–	0.341	0.420
V	0.490	0.481	0.366	–	0.434
3D	0.580	0.545	0.457	0.558	–

Ablation Experiments

Table 5.5 examines the effectiveness of the aligned attention layer, skip-connection (in attention layer), and classification loss on the NUS-WIDE-10k dataset. Our baseline is firstly modified with the addition of aligned attention layer. Including the attention shows improvement in cross-modal retrieval results.

We then ran experiments to analyze the importance of the aligned attention layer as described in (5.7) and (5.8). The improvement in attention models is attributed to the ease of training of the embedding layers. As initially reported in [136] and further demonstrated in Figure 5.5, the residual connections make the deep network easy to optimize.

Lastly, since all samples in the NUS-WIDE-10k dataset have a ground truth category, we also introduce a classification term to the loss function. This is done by training a three layer fully-connected network ($512-256-c$), where c is the number of classes. The input to this classifier network is the CVS embedding of each sample and the output is a softmax classification loss. The results show significant improvements in the retrieval scores.

Table 5.5: Mean average precision scores for cross-modal retrieval for different experiment settings on NUS-WIDE-10k dataset. AA is Aligned Attention.

Experiment Settings		NUS-WIDE-10k	
AA	Loss	Img2Txt	Text2Img
	Metric	0.439	0.585
✓	Metric	0.500	0.617
✓	Metric + Class	0.566	0.669

Attention Visualization

Figure 5.6 shows the attention vectors for image and text samples from the test set of the NUS-WIDE-10k dataset. Each plot shows the class averaged attention vector α^{it} across the embedding dimensions. We can observe peaks at different dimension for different categories. Categories such as *clouds*, *grass* and *sky* have multiple peaks distributed across the embedding dimensions. This could be attributed to the large amount of variability in these classes and presence of other other categories in these images. For example, highly overlapping categories *sky* and *cloud* have very similar curves of the respective attention vectors. Some of the other classes exhibit single peaks in the attention vectors indicating that the image and text samples are relatively easy to distinguish. Such an analysis would be very helpful in designing parameters such as embedding size of a retrieval system with a mixture of easy and hard categories.

Embedding Visualization

In order to further investigate the learned common representations, we visualize the distributions across all the modalities on the XMedia dataset. Figure 5.7 shows a t-SNE plot of CVS representation for five modalities- image, sentence, audio, video and 3D. The plot shows 1000 test samples from 20 categories. The visualization not only depicts the alignment of the five modalities but also depicts how individual categories form their own clusters. Likewise, Figure 5.8 shows the t-SNE visualization of samples from the unseen test categories from the CUB dataset. We observe natural clusters between

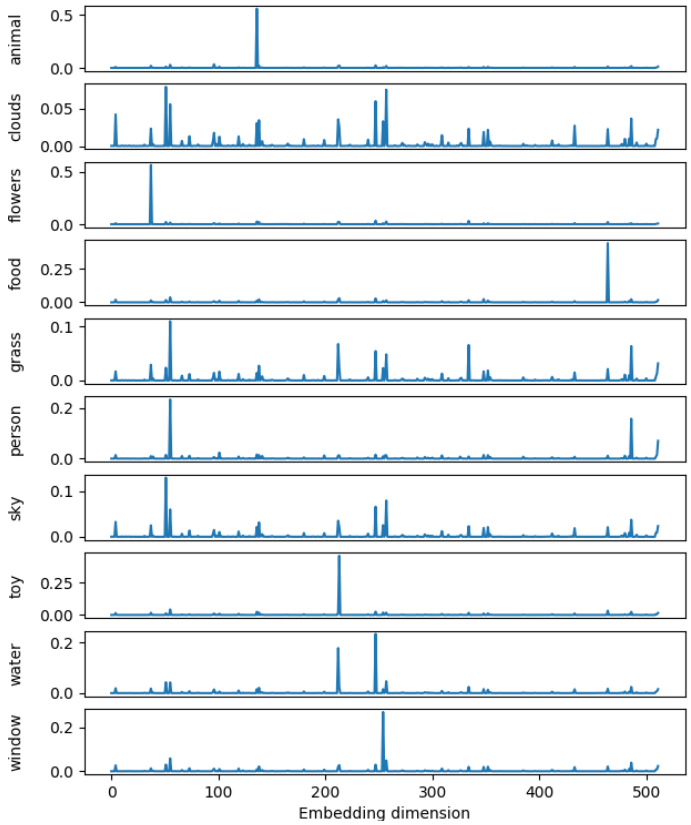


Figure 5.6: Mean attention vector (α^{is}) visualization of test samples of the ten categories in the NUS-WIDE-10k dataset.

image and sentence samples with similar semantics.

To further evaluate the quality of class clusters, we compute the Normalized Mutual Information (NMI) metric. NMI is defined by the ratio of mutual information and the average entropy of clusters and the entropy of labels [160]. In Table 5.6, we report the NMI metric for embeddings in CVS for the test samples in different datasets. CVS embeddings for all test samples are extracted. Scores are computed for individual image and text embeddings and compared with scores obtained using embeddings from both modalities jointly. Similar scores for the joint embeddings indicate that category clusters are preserved for both modalities in CVS. High scores for the two zero-shot learning datasets show the robustness of the CVS in clustering unseen categories.

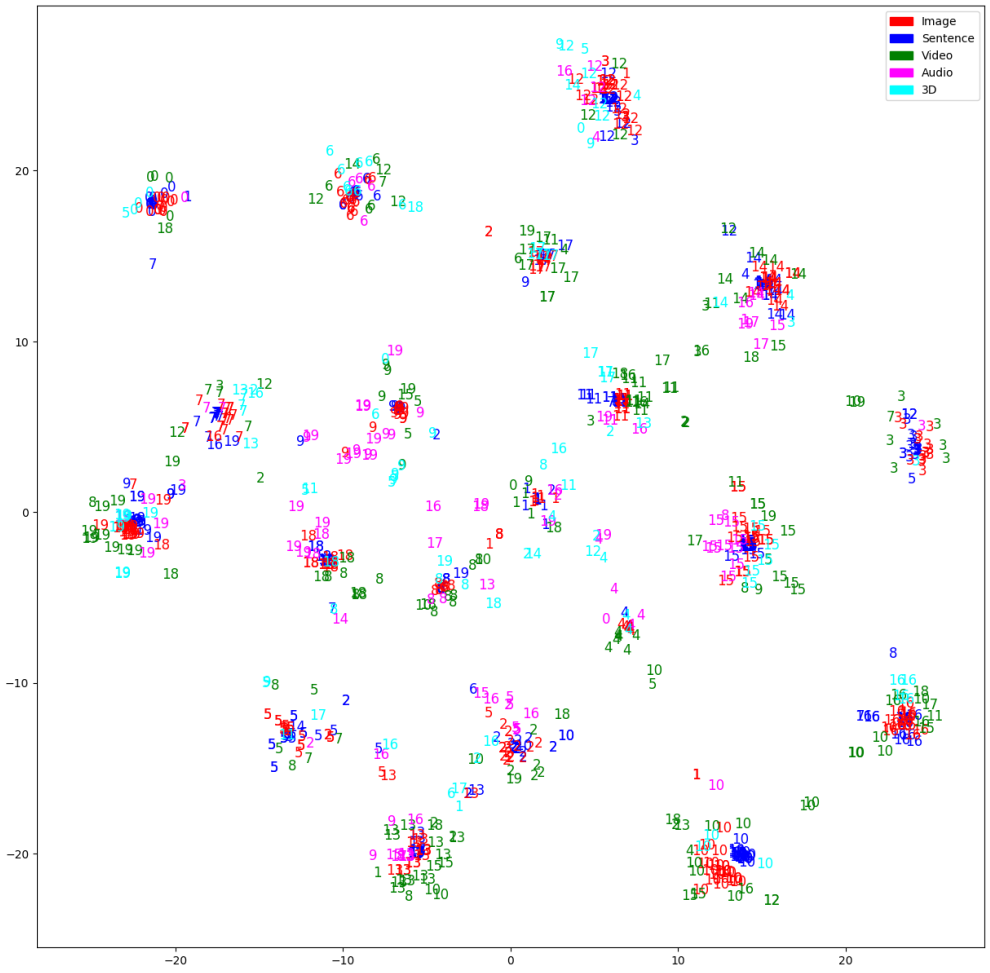


Figure 5.7: t-SNE visualization of learned CVS for XMedia dataset. Individual colors indicate different modalities and numbers denote categories. (Best viewed at 400% zoom.)

Table 5.6: NMI scores for different datasets to evaluate the quality of clusters in CVS. [1] indicates scores for unseen test categories of zero-shot learning datasets.

Dataset	Img. CVS	Sent. CVS	Joint CVS
Pascal Sent.	0.677	0.682	0.646
NUS-WIDE-10k	0.297	0.452	0.351
CUB[1]	0.741	0.660	0.659
Flowers[1]	0.600	0.628	0.609

Extensions of the CVS Model

We additionally evaluate the effectiveness of the aligned attention mechanism by evaluating the common representations learned from a retrieval model for the task of sen-

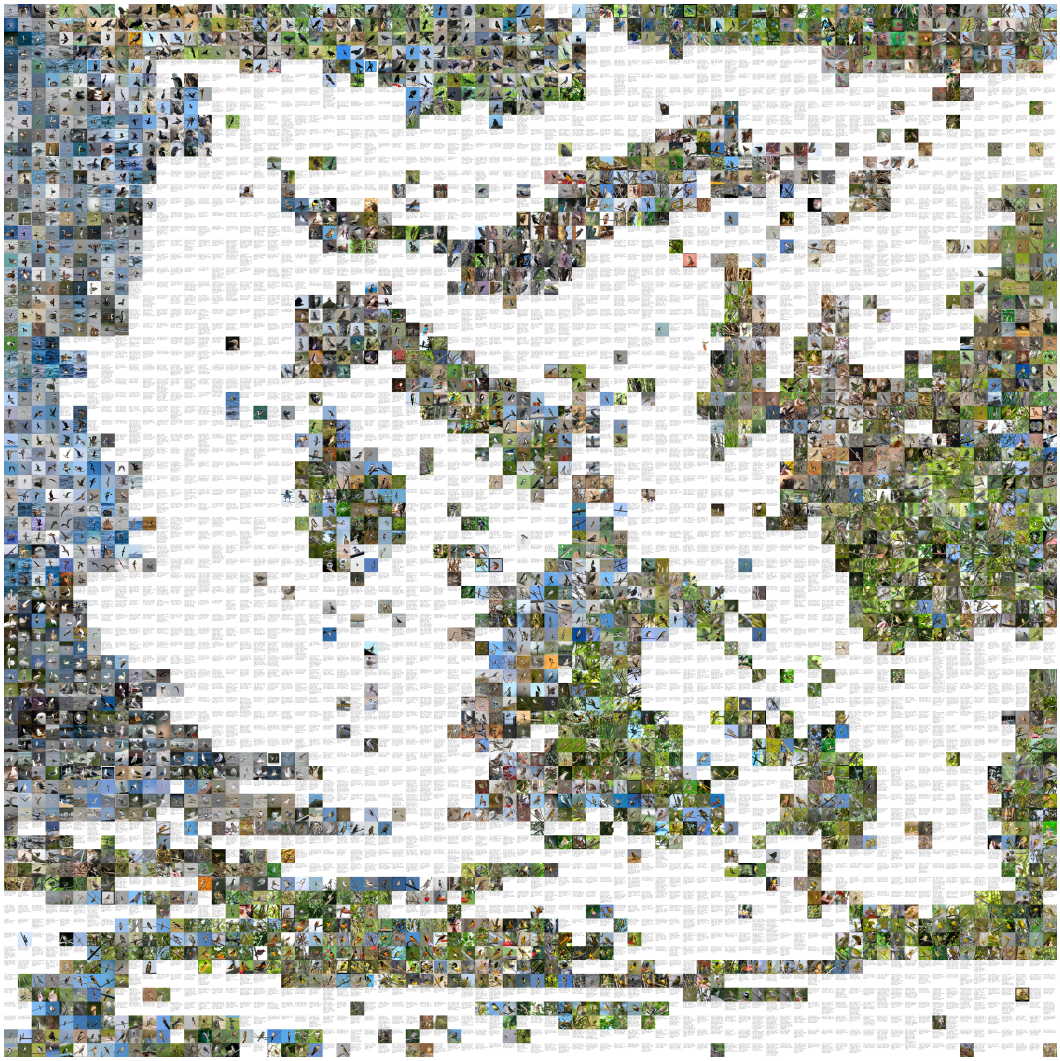


Figure 5.8: t-SNE visualization of image and sentence samples of unseen test categories from CUB dataset. (Best viewed at 1600% zoom.)

tence localization. The images from the test set of the Pascal Sentence dataset are passed through a pre-trained region proposal network [161]. The top scoring regions are aligned with a sentence to localize the sentence within the image. Sample results are shown in Figure 5.10.

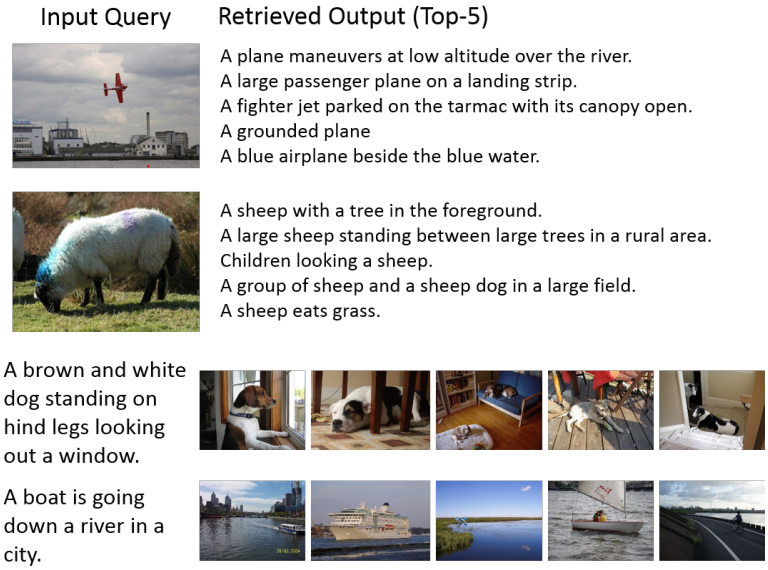


Figure 5.9: Examples of cross-modal retrieval on the Pascal Sentences dataset. Top two rows are image-to-text retrieval and bottom two rows are text-to-image retrieval. We show Top-5 retrieved samples.

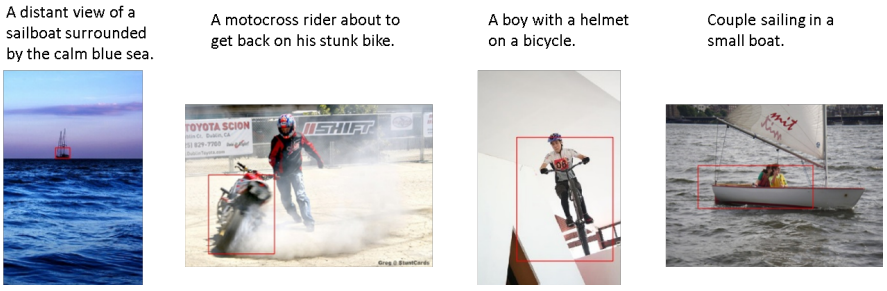


Figure 5.10: Sample results for sentence localization using the CVS model from the test set of Pascal Sentences dataset. The red box is the region aligning closest with the sentence embedding.

Chapter 6

Conclusions

In this work, models for connecting vision with natural language are developed. The problems addressed in this research are towards addressing better understanding of vision and language and their inter-dependencies. Specifically, neural network architectures that process and align the two modalities and train their parameters end-to-end on datasets of image and video captions are introduced.

In Chapter 2, a general purpose Steered Gaussian Attention Model for video understanding is introduced. Rather than using fixed training priors, video attributes are used as features along the length of the video to smartly steer attention mechanisms. When these temporal video features are bundled with a video summary vector, a semantically rich latent representation continuously feeds the captioning engine. A Gaussian parametric descriptor adds a degree of freedom to the input videos. The use of multistream hierarchical approach along with automatic boundary detection and parametric soft attention delivers state-of-the-art results on popular video captioning datasets. Through video fidelity and timing experiments, it was demonstrated that the video captioning models are robust enough to handle the power and bandwidth requirements of realistic automated surveillance systems.

In Chapter 3, a novel method for both long video summarization and annotation is introduced. Frame to frame motion, frame image quality, as well cinematographic

and consumer preference are uniquely fused together to determine interesting segments from long videos. Key frames from the most impactful segments are converted to textual annotations using an encoder-decoder recurrent neural network. Textual annotations are summarized using extractive methods where LSA, LexRank and SumBasic approaches performed best. Human evaluations of video summaries indicate promising results. Independent experiments validate both superframe cuts as well as key frame selection techniques. A key limitation is passing of incorrect superframe or key frame information to the captioning framework. A potential solution would be availability of datasets with ground truth on both key segments and associated captions/summaries.

The work in Chapter 4 inspired to develop a framework that shows the flexibility in performing cross-modal transformations. It advances the area of caption conditioned image generation by allowing the common vector space to be shared between vision and language representations. It addresses some limitations in existing studies such as one-to-one word correspondence by using the n-gram metric and conditioning on multiple semantically similar sentences. It is among the first efforts to directly tie a common vector connection space in a bidirectional visual-to-text framework by adopting image and text generative techniques. The area of image generative models has seen significant progress recently. However, evaluation techniques of the generated images are still in a nascent stage. Moreover, quality of generated images from diverse categories is limited and most current works that generate high quality images are limited to a single category.

In Chapter 5, we extend the concept of common vector space for cross-modal retrieval and zero-shot retrieval. We present a framework for learning a multi-modal common vector space. Irrespective of a sample's modality, similar concepts lie close, while dissimilar concepts lie far apart. Once in this latent representation, it is difficult to determine the original modality, making this method suitable for generic search and retrieval. Our method uses modality specific embedding functions and a new aligned attention mechanism. Weights are learned through a new multi-modal metric loss function. State-of-the-art results on image-to-text and text-to-image retrieval as well as zero-shot

learning are demonstrated across numerous datasets. We show the natural extension of our methods to five modalities- image, sentences, audio, video and 3D-models.

Overall, this thesis presents learnings through experiments for combining image and language modalities. The developed models include a video captioning framework along with an extension to summarize very long videos. We also introduced the concept of a common vector space that is shared between multiple modalities. To understand the common representations, we created models for bi-directional translation between data from image and language modalities through a generative as well as a retrieval aspect. The developed models can be extended to applications such as human-robot interaction, search & retrieval, image & video description services and video surveillance.

We have published multiple papers in various relevant areas. These include:

- Image Captioning [162, 163]
- Video Captioning [164, 165]
- Long Video Summarization [166]
- Video Activity Recognition [167]
- Very Large Deep Networks [168, 169]
- Video Redaction [170, 171]
- Graph-CNN [172]
- Sentence Paraphrasing and Summarization [173]
- Common Vector Space [174, 175, 176]
- Cross Modal Retrieval [177]

6.1 Future Work

Some potential directions for future work include −

Firstly, a current challenge is to train a generalized image generator that is capable of not only generating high quality images but also generate images from diverse categories. Most current works that generate high quality images are limited to a single category. For example, an image generator trained to synthesize only face images may generate very good face images but is limited to only the face category. In contrast, an image generator trained on multiple object categories from datasets such as MS-COCO or ImageNet is limited in terms of image quality. When trained unconditionally on the entirety of such diverse datasets, the generated images produced by current GANs have little recognizable structure, mostly producing amorphous blobs rather than recognizable objects.

Secondly, an important next step to demonstrate the value of current approaches in a more realistic settings requires learning about abstract concepts. This involves learning from very large scale unconstrained data such as information from the Internet or the physical world around us. Unsupervised and semi-supervised learning based approaches are very good at handling such kind of data. Unfortunately, current techniques are unlikely to improve learned visual representations as compared with supervised approaches trained on a sufficiently large labeled dataset like ImageNet. However, once unsupervised learning approaches mature, demonstrating the benefits will be key for their adoption. It is also important to recognize that a critical factor is that the information about the world has to be made available to the computer. This already presents many practical difficulties related to data collection and storage. Overall, this also raises the argument that computers may not reach the same level of understanding as humans have unless they can also interact with the world like we do.

Lastly, a primary challenge also lies in designing architectures that can model indefinite theories. It is insightful to note that the representations of such abstract concepts are difficult to encode in a formal language and learned by a computer. Furthermore,

for many concepts of interest, particularly in the visual world, some of the underlying factors of variation may be discrete in nature rather than continuous. However, the latent spaces typically learned through different approaches are entirely continuous. A promising approach is to allow the models to discover the internal representation of the data by its own. This is similar to the word encoding methods where the structure and relationships between words emerge as a result of optimizing an objective.

Bibliography

[1] X. Zhai, Y. Peng, and J. Xiao, "Learning cross-media joint representation with sparse and semisupervised regularization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 965–978, 2014.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, pp. 1097–1105, 2012.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[4] G. Satat, M. Tancik, O. Gupta, B. Heshmat, and R. Raskar, "Object classification through scattering media with deep learning on time resolved measurement," *Optics Express*, vol. 25, no. 15, pp. 17466–17479, 2017.

[5] B. Zhao, J. Feng, X. Wu, and S. Yan, "A survey on deep learning-based fine-grained object classification and semantic segmentation," *International Journal of Automation and Computing*, pp. 1–17, 2017.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *arXiv preprint arXiv:1708.02002*, 2017.

[8] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, and S. J. Maybank, "Salient object detection via structured matrix decomposition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 818–832, 2017.

[9] J. Long *et al.*, "Fully convolutional networks for semantic segmentation," in *CVPR*, pp. 3431–3440, 2015.

[10] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun, "Predicting deeper into the future of semantic segmentation," in *of: ICCV 2017-International Conference on Computer Vision*, p. 10, 2017.

[11] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Exploring context with deep structured models for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[12] J. Johnson *et al.*, "Densecap: Fully convolutional localization networks for dense captioning," in *CVPR*, 2016.

[13] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE CVPR*, pp. 2625–2634, 2015.

[14] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE CVPR*, pp. 3128–3137, 2015.

[15] C. Liu, J. Mao, F. Sha, and A. L. Yuille, "Attention correctness in neural image captioning.," in *AAAI*, pp. 4176–4182, 2017.

[16] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, "Semantic compositional networks for visual captioning," *arXiv preprint arXiv:1611.08002*, 2016.

[17] S. Venugopalan *et al.*, "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729*, 2014.

[18] Sutskever *et al.*, "Sequence to sequence learning with neural networks," in *NIPS*, pp. 3104–3112, 2014.

[19] S. Venugopalan *et al.*, "Sequence to sequence-video to text," in *ICCV*, pp. 4534–4542, 2015.

[20] J. Thomason *et al.*, "Integrating language and vision to generate natural language descriptions of videos in the wild.," in *Coling*, vol. 2, p. 9, 2014.

[21] R. Xu *et al.*, "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework.," in *AAAI*, pp. 2346–2352, 2015.

[22] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," *arXiv preprint arXiv:1608.06993*, 2016.

[23] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *International Journal of Computer Vision*, vol. 124, no. 2, pp. 237–254, 2017.

[24] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarizatio," *arXiv preprint arXiv:1603.03369*, 2016.

[25] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3090–3098, 2015.

[26] X. Chen and C. L. Zitnick, "Learning a recurrent visual representation for image caption generation," 2015.

[27] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, vol. 2, no. 3, p. 5, 2015.

[28] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune, "Plug & play generative networks: Conditional iterative generation of images in latent space," *arXiv preprint arXiv:1612.00005*, 2016.

[29] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[30] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, 2015.

[31] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *Advances in Neural Information Processing Systems*, pp. 3387–3395, 2016.

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[33] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

[34] M. Pedersoli, T. Lucas, C. Schmid, and J. Verbeek, "Areas of attention for image captioning," *arXiv preprint arXiv:1612.01033*, 2016.

[35] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4507–4515, 2015.

[36] P. P. et al., "Hierarchical recurrent neural encoder for video representation with application to captioning," in *CVPR*, pp. 1029–1038, 2016.

[37] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," *arXiv preprint arXiv:1510.07712*, 2015.

[38] Q. You *et al.*, "Image captioning with semantic attention," *arXiv preprint arXiv:1603.03925*, 2016.

[39] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 190–200, Association for Computational Linguistics, 2011.

[40] A. Torabi *et al.*, "Using descriptive video services to create a large data source for video annotation research," *arXiv preprint arXiv:1503.01070*, 2015.

[41] D. Tran *et al.*, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[42] Y. Pu, M. R. Min, Z. Gan, and L. Carin, "Adaptive feature abstraction for translating video to language," *arXiv preprint arXiv:1611.07837*, 2016.

[43] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[44] J. Xu *et al.*, "Msr-vtt: A large video description dataset for bridging video and language," in *CVPR*, 2016.

[45] A. Kojima, T. Tamura, and K. Fukunaga, "Natural language description of human activities from video images based on concept hierarchy of actions," *IJCV*, vol. 50, no. 2, pp. 171–184, 2002.

[46] A. Barbu *et al.*, "Video in sentences out," *arXiv preprint arXiv:1204.2742*, 2012.

[47] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, pp. 1725–1732, 2014.

[48] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, pp. 568–576, 2014.

[49] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[50] J. Dong *et al.*, "Early embedding and late reranking for video captioning," in *Proceedings of the 2016 ACM on Multimedia Conference*, pp. 1082–1086, ACM, 2016.

[51] Y. Yu, H. Ko, J. Choi, and G. Kim, "Video captioning and retrieval models with semantic attention," *arXiv preprint arXiv:1610.02947*, 2016.

[52] R. Shetty and J. Laaksonen, "Frame-and segment-level features and candidate pool evaluation for video caption generation," in *ACM on Multimedia Conf.*, pp. 1073–1076, 2016.

[53] L. A. Hendricks *et al.*, "Deep compositional captioning: Describing novel object categories without paired training data," *arXiv preprint arXiv:1511.05284*, 2015.

[54] A. Piergiovanni, C. Fan, and M. S. Ryoo, "Temporal attention filters for human activity recognition in videos," *arXiv preprint arXiv:1605.08140*, 2016.

[55] L. Baraldi, C. Grana, and R. Cucchiara, "Hierarchical boundary-aware neural encoder for video captioning," *arXiv preprint arXiv:1611.09312*, 2016.

[56] J. Xu *et al.*, "Shot boundary detection using convolutional neural networks," in *Visual Communications and Image Processing*, 2016.

[57] P. Mettes, D. C. Koelma, and C. G. Snoek, "The imagenet shuffle: Reorganized pre-training for video event detection," *arXiv preprint arXiv:1602.07119*, 2016.

[58] J. Pennington *et al.*, "Glove: Global vectors for word representation.," in *EMNLP*, vol. 14, pp. 1532–43, 2014.

[59] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014.

[60] F. Caba Heilbron *et al.*, "Activitynet: A large-scale video benchmark for human activity understanding," in *CVPR*, pp. 961–970, 2015.

[61] J. Dong, X. Li, and C. G. Snoek, "Word2visualvec: Cross-media retrieval by visual feature prediction," *arXiv preprint arXiv:1604.06838*, 2016.

[62] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, 2015.

[63] C. D. Manning *et al.*, "The stanford corenlp natural language processing toolkit.," in *ACL*, pp. 55–60, 2014.

[64] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.

[65] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[66] N. Srivastava *et al.*, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[67] *Microsoft COCO Caption Evaluation*, (accessed October 3, 2016). https://github.com/tylin/coco-caption.

[68] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318, ACL, 2002.

[69] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, vol. 29, pp. 65–72, 2005.

[70] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *CVPR*, pp. 4566–4575, 2015.

[71] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out: Proceedings of the ACL-04 Workshop*, vol. 8, Spain, 2004.

[72] V. Ramanishka *et al.*, "Multimodal video description," in *Proceedings of the ACM on Multimedia Conference*, pp. 1092–1096, 2016.

[73] F. Bellard, M. Niedermayer, *et al.*, "Ffmpeg," *Availabel from: http://ffmpeg. org*, 2012.

[74] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 1, pp. 194–201, 2012.

[75] N. Ejaz, I. Mehmood, and S. W. Baik, "Efficient visual attention based framework for extracting key frames from videos," *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 34–44, 2013.

[76] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *European conference on computer vision*, pp. 505–520, Springer, 2014.

[77] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," 2015.

[78] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence – video to text," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[79] A. Nenkova, S. Maskey, and Y. Liu, "Automatic summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts of ACL 2011*, p. 3, Association for Computational Linguistics, 2011.

[80] G. Durrett, T. Berg-Kirkpatrick, and D. Klein, "Learning-based single-document summarization with compression and anaphoricity constraints," *arXiv preprint arXiv:1603.08887*, 2016.

[81] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video," in *Proceedings of the IEEE CVPR*, pp. 2714–2721, 2013.

[82] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5179–5187, 2015.

[83] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Advances in Neural Information Processing Systems*, pp. 2069–2077, 2014.

[84] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," *arXiv preprint arXiv:1605.08110*, 2016.

[85] Y. Ke, X. Tang, and F. Jing, "The design of high-level features for photo quality assessment," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 419–426, IEEE, 2006.

[86] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Studying aesthetics in photographic images using a computational approach," in *ECCV 2006*, pp. 288–301, Springer, 2006.

[87] J. Ghosh, Y. J. Lee, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1346–1353, IEEE, 2012.

[88] R. Ptucha, D. Kloosterman, B. Mittelstaedt, and A. Loui, "Automatic image assessment from facial attributes," in *IS&T/SPIE Electronic Imaging*, pp. 90200C–90200C, International Society for Optics and Photonics, 2014.

[89] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2015.

[90] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of research and development*, vol. 2, no. 2, pp. 159–165, 1958.

[91] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264–285, 1969.

[92] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, pp. 457–479, 2004.

[93] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, "Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion," *Information Processing & Management*, vol. 43, no. 6, pp. 1606–1618, 2007.

[94] A. Haghighi and L. Vanderwende, "Exploring content models for multi-document summarization," in *Proceedings of Human Language Technologies: The 2009 An-*

nual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 362–370, Association for Computational Linguistics, 2009.

[95] T. Hirao, Y. Yoshida, M. Nishino, N. Yasuda, and M. Nagata, "Single-document summarization as a tree knapsack problem.," in *EMNLP*, vol. 13, pp. 1515–1520, 2013.

[96] A. Nenkova and L. Vanderwende, "The impact of frequency on summarization," *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*, 2005.

[97] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," 2004.

[98] G. Gkioxari and J. Malik, "Finding action tubes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 759–768, 2015.

[99] W. Wolf, "Key frame selection by motion analysis," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2, pp. 1228–1231, IEEE, 1996.

[100] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image analysis*, pp. 363–370, Springer, 2003.

[101] A. Girgensohn and J. Boreczky, "Time-constrained keyframe selection technique," in *Multimedia Computing and Systems, 1999. IEEE International Conference on*, vol. 1, pp. 756–761, IEEE, 1999.

[102] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[103] S. Yeung, A. Fathi, and L. Fei-Fei, "Videoset: Video evaluation through text," *arXiv preprint arXiv:1406.5824*, 2014.

[104] G. Guan, Z. Wang, S. Lu, J. D. Deng, and D. D. Feng, "Keypoint-based keyframe selection," *IEEE Tran on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 729–734, 2013.

[105] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

[106] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[107] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Advances in neural information processing systems*, pp. 935–943, 2013.

[108] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 689–696, 2011.

[109] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in neural information processing systems*, pp. 2222–2230, 2012.

[110] K. Sohn, W. Shang, and H. Lee, "Improved multimodal deep learning with variation of information," in *Advances in Neural Information Processing Systems*, pp. 2141–2149, 2014.

[111] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–58, 2016.

[112] K. Sohn, ""improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, 2016.

[113] A. Eisenschtat and L. Wolf, "Linking image and text with 2-way nets," in *arXiv preprint arXiv:1608.07973*, 2016.

[114] S. Wang, Yin Li, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[115] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "Starspace: Embed all the things!," *arXiv preprint arXiv:1709.03856*, 2017.

[116] A. T. L. S. . L. J. Karras, T., "Progressive growing of gans for improved quality, stability, and variation.," *arXiv preprint arXiv:1710.10196*, 2017.

[117] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," *arXiv preprint arXiv:1711.11585*, 2017.

[118] X. Liang, Z. Hu, H. Zhang, C. Gan, and E. P. Xing, "Recurrent topic-transition gan for visual paragraph generation," *arXiv preprint arXiv:1703.07022*, 2017.

[119] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, pp. 3294–3302, 2015.

[120] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Towards the imagenet-cnn of nlp: Pretraining sentence encoders with machine translation," in *Advances in Neural Information Processing Systems*, pp. 6285–6296, 2017.

[121] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425–2433, 2015.

[122] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A deep learning approach to visual question answering," *International Journal of Computer Vision*, vol. 125, no. 1-3, pp. 110–135, 2017.

[123] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, "Dense-captioning events in videos," *arXiv preprint arXiv:1705.00754*, 2017.

[124] A. Prakash, S. A. Hasan, K. Lee, V. Datla, A. Qadir, J. Liu, and O. Farri, "Neural paraphrase generation with stacked residual lstm networks," *arXiv preprint arXiv:1610.03098*, 2016.

[125] A. Gupta, A. Agarwal, P. Singh, and P. Rai, "A deep generative framework for paraphrase generation," *arXiv preprint arXiv:1709.05074*, 2017.

[126] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in Neural Information Processing Systems*, pp. 658–666, 2016.

[127] A. Dosovitskiy and T. Brox, "Inverting convolutional networks with convolutional networks," *CoRR abs/1506.02753*, 2015.

[128] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[129] P. Young *et al.*, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, 2014.

[130] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[131] I. C. Consortium *et al.*, "Specification icc. 1: 2010 (profile version 4.3. 0.0) image technology colour management–architecture, profile format, and data structure," 2010.

[132] G. B. Pawle and L. Borg, "Evolution of the icc profile connection space," in *9th Congress of the International Colour Association*, vol. 4421, pp. 446–451, International Society for Optics and Photonics, 2002.

[133] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *European Conference on Computer Vision*, pp. 451–466, Springer, 2016.

[134] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[135] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[136] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[137] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv preprint arXiv:1506.02640*, 2015.

[138] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[139] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pp. 4004–4012, IEEE, 2016.

[140] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 7–16, ACM, 2014.

[141] J. Qi, X. Huang, and Y. Peng, "Cross-media similarity metric learning with unified deep networks," *Multimedia Tools and Applications*, vol. 76, pp. 25109–25127, Dec. 2017.

[142] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, and S. Yan, "Cross-modal retrieval with cnn visual features: A new baseline," *IEEE transactions on cybernetics*, vol. 47, no. 2, pp. 449–460, 2017.

[143] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proceedings of the 2017 ACM on Multimedia Conference*, pp. 154–162, ACM, 2017.

[144] J. Qi and Y. Peng, "Cross-modal bidirectional translation via reinforcement learning.," in *IJCAI*, pp. 2630–2636, 2018.

[145] M. Dorfer, J. Schlüter, A. Vall, F. Korzeniowski, and G. Widmer, "End-to-end cross-modality retrieval with cca projections and pairwise ranking loss," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 117–128, 2018.

[146] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," *arXiv preprint arXiv:1511.06361*, 2015.

[147] K.-H. Lee, C. Xi, H. Gang, H. Houdong, and H. Xiaodong, "Stacked cross attention for image-text matching," *arXiv preprint arXiv:1803.08024*, 2018.

[148] Y. Huang, Q. Wu, and L. Wang, "Learning semantic concepts and order for image and sentence matching," *arXiv preprint arXiv:1712.02036*, 2017.

[149] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "Nus-wide: A real-world web image database from national university of singapore," in *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, (Santorini, Greece.), July 8-10, 2009.

[150] Y. Peng, X. Zhai, Y. Zhao, and X. Huang, "Semi-supervised cross-media feature learning with unified patch graph regularization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 583–596, 2016.

[151] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using amazon's mechanical turk," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 139–147, Association for Computational Linguistics, 2010.

[152] Y. Peng, X. Huang, and Y. Zhao, "An overview of cross-media retrieval: Concepts, methodologies, benchmarks and challenges," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[153] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*.

[154] X. Huang and Y. Peng, "Deep cross-media knowledge transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8837–8846, 2018.

[155] Y. Peng, J. Qi, and Y. Yuan, "Modality-specific cross-modal similarity measurement with recurrent attention network," *IEEE Transactions on Image Processing*, 2018.

[156] X. Zhang, G. Dong, Y. Du, C. Wu, Z. Luo, and C. Yang, "Collaborative subspace graph hashing for cross-modal retrieval," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pp. 213–221, ACM, 2018.

[157] Y. Peng, J. Qi, and Y. Yuan, "Cm-gans: Cross-modal generative adversarial networks for common representation learning," *arXiv preprint arXiv:1710.05106*, 2017.

[158] Y.-H. Hubert Tsai, L.-K. Huang, and R. Salakhutdinov, "Learning robust visual-semantic embeddings," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3571–3580, 2017.

[159] C. Tang, J. Lv, Y. Chen, and J. Guo, "An angle-based method for measuring the semantic similarity between visual and textual features," *Soft Computing*, pp. 1–10, 2018.

[160] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*, vol. 39. Cambridge University Press, 2008.

[161] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[162] R. M. Oruganti, S. Sah, S. Pillai, and R. Ptucha, "Image description through fusion based recurrent multi-modal learning," in *Image Processing (ICIP), 2016 IEEE International Conference on*, pp. 3613–3617, IEEE, 2016.

[163] C. Zhang, T. Nguyen, S. Sah, R. Ptucha, A. Loui, and C. Salvaggio, "Batch-normalized recurrent highway networks," in *Image Processing (ICIP), 2017 IEEE International Conference on*, IEEE, 2017.

[164] S. Sah, T. Nguyen, M. Dominguez, F. P. Such, and R. W. Ptucha, "Temporally steered gaussian attention for video understanding.," in *CVPR Workshops*, pp. 2208–2216, 2017.

[165] T. Nguyen, S. Sah, and R. Ptucha, "Multistream hierarchical boundary network for video captioning," in *2017 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, pp. 1–5, IEEE, 2017.

[166] S. Sah, S. Kulhare, A. Gray, S. Venugopalan, E. Prud'Hommeaux, and R. Ptucha, "Semantic text summarization of long videos," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pp. 989–997, IEEE, 2017.

[167] S. Kulhare, S. Sah, S. Pillai, and R. Ptucha, "Key frame extraction for salient activity recognition," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 835–840, IEEE, 2016.

[168] S. Chennupati, S. Sah, S. Nooka, and R. Ptucha, "Hierarchical decomposition of large deep networks," *Electronic Imaging*, vol. 2016, no. 19, pp. 1–6, 2016.

[169] S. P. Nooka, S. Chennupati, K. Veerabhadra, S. Sah, and R. Ptucha, "Adaptive hierarchical classification networks," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 3578–3583, IEEE, 2016.

[170] S. Sah, A. Shringi, R. Ptucha, A. M. Burry, and R. P. Loce, "Video redaction: a survey and comparison of enabling technologies," *Journal of Electronic Imaging*, vol. 26, no. 5, p. 051406, 2017.

[171] S. Sah, R. Longman, A. Shringi, R. Loce, M. Rabbani, and R. Ptucha, "Detection without recognition for redaction,"

[172] F. P. Such, S. Sah, M. A. Dominguez, S. Pillai, C. Zhang, A. Michael, N. D. Cahill, and R. Ptucha, "Robust spatial filtering with graph convolutional neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 884–896, 2017.

[173] C. Zhang, S. Sah, T. Nguyen, P. Dheeraj, A. Loui, C. Salvaggio, and R. Ptucha, "Semantic sentence embeddings for paraphrasing and text summarization," in *Global Conference on Signal and Information Processing (GlobalSIP)*, 2017.

[174] S. Sah, C. Zhang, T. Nguyen, D. K. Peri, A. Shringi, and R. Ptucha, "Vector learning for cross domain representations," *arXiv preprint arXiv:1809.10312*, 2018.

[175] S. Sah, D. Peri, A. Shringi, C. Zhang, M. Dominguez, A. Savakis, and R. Ptucha, "Semantically invariant text-to-image generation," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 3783–3787, IEEE, 2018.

[176] S. Sah, A. Shringi, D. Peri, J. Hamilton, A. Savakis, and R. Ptucha, "Multimodal reconstruction using vector representation," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 3763–3767, IEEE, 2018.

[177] S. Sah, S. Gopalakrishnan, and R. Ptucha, "Cross modal retrieval using common vector space,"

[178] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *CVPR*, 2016.

[179] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *CVPR*, 2016.

[180] Pan *et al.*, "Hierarchical recurrent neural encoder for video representation with application to captioning," *arXiv preprint arXiv:1511.03476*, 2015.

[181] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for movie description," in *CVPR*, 2015.

[182] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, June 2015.

[183] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[184] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, *et al.*, "From captions to visual concepts and back," in *CVPR*, pp. 1473–1482, 2015.

[185] N. Krishnamoorthy, G. Malkarnenkar, R. J. Mooney, K. Saenko, and S. Guadarrama, "Generating natural-language video descriptions using text-mined knowledge.," in *AAAI*, vol. 1, p. 2, 2013.

[186] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, pp. 1–9, 2015.

[187] J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber, "Recurrent highway networks," *arXiv preprint arXiv:1607.03474*, 2016.

[188] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[189] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[190] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[191] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, ACM, 2008.

[192] H. Amiri, P. Resnik, J. Boyd-Graber, and H. Daumé III, "Learning text pair similarity with context-sensitive autoencoders," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1882–1892, 2016.

[193] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678, ACM, 2014.

[194] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *arXiv preprint arXiv:1612.03242*, 2016.

[195] J. Choi, T.-H. Oh, and I. S. Kweon, "Textually customized video summaries," *arXiv preprint arXiv:1702.01528*, 2017.

[196] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[197] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," *arXiv preprint arXiv:1711.11585*, 2017.

[198] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[199] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[200] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[201] R. Vedantam, C. L. Zitnick, and D. Parikh, "Collecting image description datasets using crowdsourcing," *arXiv preprint arXiv:1411.3041*, 2014.

[202] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[203] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.

[204] X. Xu, L. He, H. Lu, L. Gao, and Y. Ji, "Deep adversarial metric learning for cross-modal retrieval," *World Wide Web*, pp. 1–16, 2018.

[205] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," *CoRR*, 2017.

[206] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 251–260, ACM, 2010.

[207] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

[208] S. Li, T. Xiao, H. Li, W. Yang, and X. Wang, "Identity-aware textual-visual matching with latent co-attention," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 1908–1917, IEEE, 2017.

[209] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[210] T. X. Elyor Kodirov and S. Gong, "Semantic Autoencoder for Zero-shot Learning," 2017.

[211] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5327–5336, 2016.

[212] L. Zhang, T. Xiang, S. Gong, *et al.*, "Learning a deep embedding model for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE CVPR, 2017.

[213] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning the good, the bad and the ugly," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3077–3086, 2017.

[214] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2927–2936, 2015.

[215] X. Huang, Y. Peng, and M. Yuan, "Mhtn: Modal-adversarial hybrid transfer network for cross-modal retrieval," *arXiv preprint arXiv:1708.04308*, 2017.